W3School CSS 参考手册

wizardforcel

Published with GitBook



目錄

W3School CSS 参考手册

来源：CSS 参考手册

整理：飞龙

CSS3 动画属性（Animation）

属性	描述	CSS
@keyframes	规定动画。	3
animation	所有动画属性的简写属性，除了 animation-play-state 属性。	3
animation-name	规定 @keyframes 动画的名称。	3
animation-duration	规定动画完成一个周期所花费的秒或毫秒。	3
animation-timing-function	规定动画的速度曲线。	3
animation-delay	规定动画何时开始。	3
animation-iteration-count	规定动画被播放的次数。	3
animation-direction	规定动画是否在下一周期逆向地播放。	3
animation-play-state	规定动画是否正在运行或暂停。	3
animation-fill-mode	规定对象动画时间之外的状态。	3

CSS3 @keyframes 规则

实例

使 div 元素匀速向下移动：

```
@keyframes mymove
{
from {top:0px;}
to {top:200px;}
}

@-moz-keyframes mymove /* Firefox */
{
from {top:0px;}
to {top:200px;}
}

@-webkit-keyframes mymove /* Safari 和 Chrome */
{
from {top:0px;}
to {top:200px;}
}

@-o-keyframes mymove /* Opera */
{
from {top:0px;}
to {top:200px;}
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

目前浏览器都不支持 @keyframes 规则。

Firefox 支持替代的 @-moz-keyframes 规则。

Opera 支持替代的 @-o-keyframes 规则。

Safari 和 Chrome 支持替代的 @-webkit-keyframes 规则。

定义和用法

通过 @keyframes 规则，您能够创建动画。

创建动画的原理是，将一套 CSS 样式逐渐变化为另一套样式。

在动画过程中，您能够多次改变这套 CSS 样式。

以百分比来规定改变发生的时间，或者通过关键词 "from" 和 "to"，等价于 0% 和 100%。

0% 是动画的开始时间，100% 动画的结束时间。

为了获得最佳的浏览器支持，您应该始终定义 0% 和 100% 选择器。

注释：请使用动画属性来控制动画的外观，同时将动画与选择器绑定。

语法

```
@keyframes _animationname_ {_keyframes-selector_ {_css-styles_;}}
```

值	描述
animationname	必需。定义动画的名称。
keyframes-selector	

必需。动画时长的百分比。

合法的值：

- 0-100%
- from（与 0% 相同）
- to（与 100% 相同）

  | | *css-styles* | 必需。一个或多个合法的 CSS 样式属性。 |

亲自试一试 - 实例

实例 1

在一个动画中添加多个 keyframe 选择器：

```
@keyframes mymove
{
0%   {top:0px;}
25%  {top:200px;}
50%  {top:100px;}
75%  {top:200px;}
100% {top:0px;}
}

@-moz-keyframes mymove /* Firefox */
{
0%   {top:0px;}
25%  {top:200px;}
50%  {top:100px;}
75%  {top:200px;}
100% {top:0px;}
}

@-webkit-keyframes mymove /* Safari 和 Chrome */
{
0%   {top:0px;}
25%  {top:200px;}
50%  {top:100px;}
75%  {top:200px;}
100% {top:0px;}
}

@-o-keyframes mymove /* Opera */
{
0%   {top:0px;}
25%  {top:200px;}
50%  {top:100px;}
75%  {top:200px;}
100% {top:0px;}
}
```

实例 2

在一个动画中改变多个 CSS 样式：

```
@keyframes mymove
{
0%   {top:0px; background:red; width:100px;}
100% {top:200px; background:yellow; width:300px;}
}

@-moz-keyframes mymove /* Firefox */
{
0%   {top:0px; background:red; width:100px;}
100% {top:200px; background:yellow; width:300px;}
}

@-webkit-keyframes mymove /* Safari 和 Chrome */
{
0%   {top:0px; background:red; width:100px;}
100% {top:200px; background:yellow; width:300px;}
}

@-o-keyframes mymove /* Opera */
{
0%   {top:0px; background:red; width:100px;}
100% {top:200px; background:yellow; width:300px;}
}
```

实例 3

带有多个 CSS 样式的多个 keyframe 选择器：

```
@keyframes mymove
{
0%   {top:0px; left:0px; background:red;}
25%  {top:0px; left:100px; background:blue;}
50%  {top:100px; left:100px; background:yellow;}
75%  {top:100px; left:0px; background:green;}
100% {top:0px; left:0px; background:red;}
}

@-moz-keyframes mymove /* Firefox */
{
0%   {top:0px; left:0px; background:red;}
25%  {top:0px; left:100px; background:blue;}
50%  {top:100px; left:100px; background:yellow;}
75%  {top:100px; left:0px; background:green;}
100% {top:0px; left:0px; background:red;}
}

@-webkit-keyframes mymove /* Safari and Chrome */
{
0%   {top:0px; left:0px; background:red;}
25%  {top:0px; left:100px; background:blue;}
50%  {top:100px; left:100px; background:yellow;}
75%  {top:100px; left:0px; background:green;}
100% {top:0px; left:0px; background:red;}
}

@-o-keyframes mymove /* Opera */
{
0%   {top:0px; left:0px; background:red;}
25%  {top:0px; left:100px; background:blue;}
50%  {top:100px; left:100px; background:yellow;}
75%  {top:100px; left:0px; background:green;}
100% {top:0px; left:0px; background:red;}
}
```

相关页面

CSS3 教程：CSS3 动画

CSS3 animation 属性

实例

使用简写属性，将动画与 div 元素绑定：

```
div
{
animation:mymove 5s infinite;
-webkit-animation:mymove 5s infinite; /* Safari 和 Chrome */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox 以及 Opera 支持 animation 属性。

Safari 和 Chrome 支持替代的 -webkit-animation 属性。

注释：Internet Explorer 9 以及更早的版本不支持 animation 属性。

定义和用法

animation 属性是一个简写属性，用于设置六个动画属性：

- animation-name
- animation-duration
- animation-timing-function
- animation-delay
- animation-iteration-count
- animation-direction

注释：请始终规定 animation-duration 属性，否则时长为 0，就不会播放动画了。

默认值：	none 0 ease 0 1 normal
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.animation="mymove 5s infinite"

语法

```
animation: _name_ _duration_ _timing-function_ _delay_ _iteration-count_ _direction_;
```

值	描述
animation-name	规定需要绑定到选择器的 keyframe 名称。。
animation-duration	规定完成动画所花费的时间，以秒或毫秒计。
animation-timing-function	规定动画的速度曲线。
animation-delay	规定在动画开始之前的延迟。
animation-iteration-count	规定动画应该播放的次数。
animation-direction	规定是否应该轮流反向播放动画。

相关页面

CSS3 教程：CSS3 动画

CSS3 animation-name 属性

实例

为 @keyframes 动画规定一个名称：

```
div
{
animation:mymove 5s infinite;
-webkit-animation:mymove 5s infinite; /* Safari 和 Chrome */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox 以及 Opera 支持 animation-name 属性。

Safari 和 Chrome 支持替代的 -webkit-animation-name 属性。

注释：Internet Explorer 9 以及更早的版本不支持 animation-name 属性。

定义和用法

animation-name 属性为 @keyframes 动画规定名称。

注释：请始终规定 animation-duration 属性，否则时长为 0，就不会播放动画了。

默认值：	none
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.animationName="mymove"

语法

```
animation-name: _keyframename_|none;
```

值	描述
keyframename	规定需要绑定到选择器的 keyframe 的名称。
none	规定无动画效果（可用于覆盖来自级联的动画）。

相关页面

CSS3 教程：CSS3 动画

CSS3 animation-duration 属性

实例

为 @keyframes 动画规定一个名称：

```
div
{
animation-duration:2s;
-webkit-animation-duration:2s; /* Safari 和 Chrome */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox 以及 Opera 支持 animation-duration 属性。

Safari 和 Chrome 支持替代的 -webkit-animation-duration 属性。

注释：Internet Explorer 9 以及更早的版本不支持 animation-duration 属性。

定义和用法

animation-duration 属性定义动画完成一个周期所需要的时间，以秒或毫秒计。

默认值：	0
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.animationDuration="3s"

语法

```
animation-duration: _time_;
```

值	描述
time	规定完成动画所花费的时间。默认值是 0，意味着没有动画效果。

相关页面

CSS3 教程：CSS3 动画

CSS3 animation-timing-function 属性

实例

从开头到结尾以相同的速度来播放动画：

```
div
{
animation-timing-function:2s;
-webkit-animation-timing-function:2s; /* Safari 和 Chrome */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox 以及 Opera 支持 animation-timing-function 属性。

Safari 和 Chrome 支持替代的 -webkit-animation-timing-function 属性。

注释：Internet Explorer 9 以及更早的版本不支持 animation-timing-function 属性。

定义和用法

animation-timing-function 规定动画的速度曲线。

速度曲线定义动画从一套 CSS 样式变为另一套所用的时间。

速度曲线用于使变化更为平滑。

默认值：	ease
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.animationTimingFunction="linear"

语法

```
animation-timing-function: _value_;
```

animation-timing-function 使用名为三次贝塞尔（Cubic Bezier）函数的数学函数，来生成速度曲线。您能够在该函数中使用自己的值，也可以预定义的值：

值	描述	测试
linear	动画从头到尾的速度是相同的。	测试
ease	默认。动画以低速开始，然后加快，在结束前变慢。	测试
ease-in	动画以低速开始。	测试
ease-out	动画以低速结束。	测试
ease-in-out	动画以低速开始和结束。	测试
cubic-bezier(*n*,*n*,*n*,*n*)	在 cubic-bezier 函数中自己的值。可能的值是从 0 到 1 的数值。	

提示：请试着在下面的“亲自试一试”功能中使用不同的值。

亲自试一试 - 实例

实例 1

为了更好地理解不同的定时函数值，这里提供了设置五个不同值的五个不同的 div 元素：

```
/* W3C 和 Opera: */
#div1 {animation-timing-function: linear;}
#div2 {animation-timing-function: ease;}
#div3 {animation-timing-function: ease-in;}
#div4 {animation-timing-function: ease-out;}
#div5 {animation-timing-function: ease-in-out;}
/* Firefox: */
#div1 {-moz-animation-timing-function: linear;}
#div2 {-moz-animation-timing-function: ease;}
#div3 {-moz-animation-timing-function: ease-in;}
#div4 {-moz-animation-timing-function: ease-out;}
#div5 {-moz-animation-timing-function: ease-in-out;}
/* Safari 和 Chrome: */
#div1 {-webkit-animation-timing-function: linear;}
#div2 {-webkit-animation-timing-function: ease;}
#div3 {-webkit-animation-timing-function: ease-in;}
#div4 {-webkit-animation-timing-function: ease-out;}
#div5 {-webkit-animation-timing-function: ease-in-out;}
```

实例 2

与上例相同，但是通过 cubic-bezier 函数来定义速度曲线：

```
/* W3C 和 Opera: */
#div1 {animation-timing-function: cubic-bezier(0,0,1,1);}
#div2 {animation-timing-function: cubic-bezier(0.25,0.1,0.25,1);}
#div3 {animation-timing-function: cubic-bezier(0.42,0,1,1);}
#div4 {animation-timing-function: cubic-bezier(0,0,0.58,1);}
#div5 {animation-timing-function: cubic-bezier(0.42,0,0.58,1);}
/* Firefox: */
#div1 {-moz-animation-timing-function: cubic-bezier(0,0,1,1);}
#div2 {-moz-animation-timing-function: cubic-bezier(0.25,0.1,0.25,1);}
#div3 {-moz-animation-timing-function: cubic-bezier(0.42,0,1,1);}
#div4 {-moz-animation-timing-function: cubic-bezier(0,0,0.58,1);}
#div5 {-moz-animation-timing-function: cubic-bezier(0.42,0,0.58,1);}
/* Safari 和 Chrome: */
#div1 {-webkit-animation-timing-function: cubic-bezier(0,0,1,1);}
#div2 {-webkit-animation-timing-function: cubic-bezier(0.25,0.1,0.25,1);}
#div3 {-webkit-animation-timing-function: cubic-bezier(0.42,0,1,1);}
#div4 {-webkit-animation-timing-function: cubic-bezier(0,0,0.58,1);}
#div5 {-webkit-animation-timing-function: cubic-bezier(0.42,0,0.58,1);}
```

相关页面

CSS3 教程：CSS3 动画

CSS3 animation-delay 属性

实例

等待两秒，然后开始动画：

```
div
{
animation-delay:2s;
-webkit-animation-delay:2s; /* Safari 和 Chrome */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox 以及 Opera 支持 animation-delay 属性。

Safari 和 Chrome 支持替代的 -webkit-animation-delay 属性。

注释：Internet Explorer 9 以及更早的版本不支持 animation-delay 属性。

定义和用法

animation-delay 属性定义动画何时开始。

animation-delay 值以秒或毫秒计。

提示：允许负值，-2s 使动画马上开始，但跳过 2 秒进入动画。

默认值：	0
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.animationDelay="2s"
---	---

语法

```
animation-delay: _time_;
```

值	描述	测试
time	可选。定义动画开始前等待的时间，以秒或毫秒计。默认值是 0。	测试

亲自试一试 - 实例

负值，请注意动画跳过 2 秒进入动画周期：

```
animation-delay: -2s /* W3C 和 Opera */
-moz-animation-delay: -2s /* Firefox */
-webkit-animation-delay: -2s /* Safari 和 Chrome */
```

相关页面

CSS3 教程：CSS3 动画

CSS3 animation-iteration-count 属性

实例

播放动画三次：

```
div
{
animation-iteration-count:3;
-webkit-animation-iteration-count:3; /* Safari 和 Chrome */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox 以及 Opera 支持 animation-iteration-count 属性。

Safari 和 Chrome 支持替代的 -webkit-animation-iteration-count 属性。

注释：Internet Explorer 9 以及更早的版本不支持 animation-iteration-count 属性。

定义和用法

animation-iteration-count 属性定义动画的播放次数。

默认值：	1
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.animationIterationCount=3

语法

```
animation-iteration-count: _n_|infinite;
```

值	描述	测试
n	定义动画播放次数的数值。	测试
infinite	规定动画应该无限次播放。	测试

相关页面

CSS3 教程：CSS3 动画

CSS3 animation-direction 属性

实例

暂停动画：

```
div
{
animation-direction:alternate;
-webkit-animation-direction:alternate; /* Safari 和 Chrome */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox 以及 Opera 支持 animation-direction 属性。

Safari 和 Chrome 支持替代的 -webkit-animation-direction 属性。

注释：Internet Explorer 9 以及更早的版本不支持 animation-direction 属性。

定义和用法

animation-direction 属性定义是否应该轮流反向播放动画。

如果 animation-direction 值是 "alternate"，则动画会在奇数次数（1、3、5 等等）正常播放，而在偶数次数（2、4、6 等等）向后播放。

注释：如果把动画设置为只播放一次，则该属性没有效果。

默认值：	normal
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.animationDirection="alternate"
---	---

语法

```
animation-direction: normal|alternate;
```

值	描述	测试
normal	默认值。动画应该正常播放。	测试
alternate	动画应该轮流反向播放。	测试

相关页面

CSS3 教程：CSS3 动画

CSS3 animation-play-state 属性

实例

暂停动画：

```
div
{
animation-play-state:paused;
-webkit-animation-play-state:paused; /* Safari 和 Chrome */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox 以及 Opera 支持 animation-play-state 属性。

Safari 和 Chrome 支持替代的 -webkit-animation-play-state 属性。

注释：Internet Explorer 9 以及更早的版本不支持 animation-play-state 属性。

定义和用法

animation-play-state 属性规定动画正在运行还是暂停。

注释：您可以在 JavaScript 中使用该属性，这样就能在播放过程中暂停动画。

默认值：	running
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.animationPlayState="paused"
---	---

语法

```
animation-play-state: paused|running;
```

值	描述	测试
paused	规定动画已暂停。	测试
running	规定动画正在播放。	测试

相关页面

CSS3 教程：CSS3 动画

CSS3 animation-fill-mode 属性

实例

为 h1 元素规定填充模式：

```
h1
  {
  animation-fill-mode: forwards;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox 以及 Opera 支持 animation-fill-mode 属性。

Safari 和 Chrome 支持替代的 -webkit-animation-fill-mode 属性。

注释：Internet Explorer 9 以及更早的版本不支持 animation-fill-mode 属性。

定义和用法

animation-fill-mode 属性规定动画在播放之前或之后，其动画效果是否可见。

注释：其属性值是由逗号分隔的一个或多个填充模式关键词。

默认值：	none
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.animationFillMode=none

语法

```
animation-fill-mode : none | forwards | backwards | both;
```

值	描述
none	不改变默认行为。
forwards	当动画完成后，保持最后一个属性值（在最后一个关键帧中定义）。
backwards	在 animation-delay 所指定的一段时间内，在动画显示之前，应用开始属性值（在第一个关键帧中定义）。
both	向前和向后填充模式都被应用。

相关页面

CSS3 教程：CSS3 动画

CSS 背景属性（Background）

属性	描述	CSS
background	在一个声明中设置所有的背景属性。	1
background-attachment	设置背景图像是否固定或者随着页面的其余部分滚动。	1
background-color	设置元素的背景颜色。	1
background-image	设置元素的背景图像。	1
background-position	设置背景图像的开始位置。	1
background-repeat	设置是否及如何重复背景图像。	1
background-clip	规定背景的绘制区域。	3
background-origin	规定背景图片的定位区域。	3
background-size	规定背景图片的尺寸。	3

CSS background 属性

实例

如何在一个声明中设置所有背景属性：

```
body
  {
  background: #00FF00 url(bgimage.gif) no-repeat fixed top;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 background 属性。

注释：IE8 以及更早的浏览器不支持一个元素多个背景图像。

注释：IE7 以及更早的浏览器不支持 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

background 简写属性在一个声明中设置所有的背景属性。

可以设置如下属性：

- background-color
- background-position
- background-size
- background-repeat
- background-origin
- background-clip
- background-attachment
- background-image

如果不设置其中的某个值，也不会出问题，比如 background:#ff0000 url('smiley.gif'); 也是允许的。

通常建议使用这个属性，而不是分别使用单个属性，因为这个属性在较老的浏览器中能够得到更好的支持，而且需要键入的字母也更少。

默认值：	***not specified***
继承性：	no
版本：	CSS1 + CSS3
JavaScript 语法：	*object*.style.background="white url(paper.gif) repeat-y"

可能的值

值	描述	CSS
background-color	规定要使用的背景颜色。	1
background-position	规定背景图像的位置。	1
background-size	规定背景图片的尺寸。	3
background-repeat	规定如何重复背景图像。	1
background-origin	规定背景图片的定位区域。	3
background-clip	规定背景的绘制区域。	3
background-attachment	规定背景图像是否固定或者随着页面的其余部分滚动。	1
background-image	规定要使用的背景图像。	1
inherit	规定应该从父元素继承 background 属性的设置。	1

亲自试一试 - 实例

所有背景属性在一个声明之中

本例演示如何使用简写属性来将所有背景属性设置在一个声明之中。

```
<html>
<head>
<style type="text/css">
body
{
background: #ff0000 url(/i/eg_bg_03.gif) no-repeat fixed center;
}
</style>
</head>

<body>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
<p>这是一些文本。</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 背景

CSS3 教程：CSS3 背景

HTML DOM 参考手册：background 属性

CSS background-attachment 属性

实例

如何设置固定的背景图像：

```
body
  {
  background-image: url(bgimage.gif);
  background-attachment: fixed;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 background-attachment 属性。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

background-attachment 属性设置背景图像是否固定或者随着页面的其余部分滚动。

默认值：	scroll
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.backgroundAttachment="fixed"

可能的值

值	描述
scroll	默认值。背景图像会随着页面其余部分的滚动而移动。
fixed	当页面的其余部分滚动时，背景图像不会移动。
inherit	规定应该从父元素继承 background-attachment 属性的设置。

TIY 实例

如何设置固定的背景图像

本例演示如何设置固定的背景图像。图像不会随着页面的其他部分滚动。

```
<html>
<head>
<style type="text/css">
body
{
background-image:url(/i/eg_bg_02.gif);
background-repeat:no-repeat;
background-attachment:fixed
}
</style>
</head>

<body>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
<p>图像不会随页面的其余部分滚动。</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 背景

HTML DOM 参考手册：backgroundAttachment 属性

CSS background-color 属性

实例

```
body
  {
  background-color:yellow;
  }
h1
  {
  background-color:#00ff00;
  }
p
  {
  background-color:rgb(255,0,255);
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 background-color 属性。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

background-color 属性设置元素的背景颜色。

元素背景的范围

background-color 属性为元素设置一种纯色。这种颜色会填充元素的内容、内边距和边框区域，扩展到元素边框的外边界（但不包括外边距）。如果边框有透明部分（如虚线边框），会透过这些透明部分显示出背景色。

transparent 值

尽管在大多数情况下，没有必要使用 transparent。不过如果您不希望某元素拥有背景色，同时又不希望用户对浏览器的颜色设置影响到您的设计，那么设置 transparent 值还是有必要的。

默认值：	**transparent**
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.backgroundColor="#00FF00"

可能的值

值	描述
color_name	规定颜色值为颜色名称的背景颜色（比如 red）。
hex_number	规定颜色值为十六进制值的背景颜色（比如 #ff0000）。
rgb_number	规定颜色值为 rgb 代码的背景颜色（比如 rgb(255,0,0)）。
transparent	默认。背景颜色为透明。
inherit	规定应该从父元素继承 background-color 属性的设置。

TIY 实例

设置背景颜色

本例演示如何为元素设置背景颜色。

```
<html>
<head>

<style type="text/css">

body {background-color: yellow}
h1 {background-color: #00ff00}
h2 {background-color: transparent}
p {background-color: rgb(250,0,255)}

p.no2 {background-color: gray; padding: 20px;}

</style>

</head>

<body>

<h1>这是标题 1</h1>
<h2>这是标题 2</h2>
<p>这是段落</p>
<p class="no2">这个段落设置了内边距。</p>

</body>
</html>
```

设置文本的背景颜色

本例颜色如何设置部分文本的背景颜色。

```
<html>
<head>
<style type="text/css">
span.highlight
{
background-color:yellow
}
</style>
</head>

<body>
<p>
<span class="highlight">这是文本。</span> 这是文本。 这是文本。 这是文本。 这是文本。 这是文本。 这
</p>
</body>
</html>
```

相关页面

CSS 教程：CSS 背景

HTML DOM 参考手册：backgroundColor 属性

CSS background-image 属性

实例

```
body
  {
  background-image: url(bgimage.gif);
  background-color: #000000;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 background-image 属性。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

background-image 属性为元素设置背景图像。

元素的背景占据了元素的全部尺寸，包括内边距和边框，但不包括外边距。

默认地，背景图像位于元素的左上角，并在水平和垂直方向上重复。

提示：请设置一种可用的背景颜色，这样的话，假如背景图像不可用，页面也可获得良好的视觉效果。

详细说明

background-image 属性会在元素的背景中设置一个图像。

根据 background-repeat 属性的值，图像可以无限平铺、沿着某个轴（x 轴或 y 轴）平铺，或者根本不平铺。

初始背景图像（原图像）根据 background-position 属性的值放置。

默认值：	**none**
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.backgroundImage="url(stars.gif)"

可能的值

值	描述
url('*URL*')	指向图像的路径。
none	默认值。不显示背景图像。
inherit	规定应该从父元素继承 background-image 属性的设置。

TIY 实例

将图像设置为背景

本例演示如何将图像设置为背景。

```
<html>
<head>

<style type="text/css">
body {background-image:url(/i/eg_bg_04.gif);}
</style>

</head>

<body></body>

</html>
```

相关页面

CSS 教程：CSS 背景

HTML DOM 参考手册：backgroundImage 属性

CSS background-position 属性

实例

如何定位背景图像：

```
body
{
background-image:url('bgimage.gif');
background-repeat:no-repeat;
background-attachment:fixed;
background-position:center;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 background-position 属性。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

background-position 属性设置背景图像的起始位置。

这个属性设置背景原图像（由 background-image 定义）的位置，背景图像如果要重复，将从这一点开始。

提示：您需要把 background-attachment 属性设置为 "fixed"，才能保证该属性在 Firefox 和 Opera 中正常工作。

默认值：	0% 0%
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.backgroundPosition="center"

可能的值

值	描述

- top left
- top center
- top right
- center left
- center center
- center right
- bottom left
- bottom center
- bottom right

|

如果您仅规定了一个关键词，那么第二个值将是"center"。

默认值：0% 0%。

| | x% y% |

第一个值是水平位置，第二个值是垂直位置。

左上角是 0% 0%。右下角是 100% 100%。

如果您仅规定了一个值，另一个值将是 50%。

| | xpos ypos |

第一个值是水平位置，第二个值是垂直位置。

左上角是 0 0。单位是像素 (0px 0px) 或任何其他的 CSS 单位。

如果您仅规定了一个值，另一个值将是50%。

您可以混合使用 % 和 position 值。

|

TIY 实例

如何放置背景图像

本例演示如何在页面上放置背景图像。

```
<html>
<head>
<style type="text/css">
body
{
  background-image:url('/i/eg_bg_03.gif');
  background-repeat:no-repeat;
  background-attachment:fixed;
  background-position:center;
}
</style>
</head>

<body>
<body>
<p><b>提示：</b>您需要把 background-attachment 属性设置为 "fixed"，才能保证该属性在 Firefox 和 O
</body>
</body>
</html>
```

如何使用%来定位背景图像

本例演示如何使用百分比来在页面上定位背景图像。

```
<html>
<head>
<style type="text/css">
body
{
background-image: url('/i/eg_bg_03.gif');
background-repeat: no-repeat;
background-attachment:fixed;
background-position: 30% 20%;
}
</style>
</head>

<body>
<p><b>注释：</b>为了在 Mozilla 中实现此效果，background-attachment 属性必须设置为 "fixed"。</p>
</body>
</html>
```

如何使用像素来定位背景图像

本例演示如何使用像素来在页面上定位背景图像。

```
<html>
<head>
<style type="text/css">
body
{
background-image: url('/i/eg_bg_03.gif');
background-repeat: no-repeat;
background-attachment:fixed;
background-position: 50px 100px;
}
</style>
</head>

<body>
<p><b>注释：</b>为了在 Mozilla 中实现此效果，background-attachment 属性必须设置为 "fixed"。</p>
</body>
</html>
```

相关页面

CSS 教程：CSS 背景

CSS 参考手册：background-image 属性

HTML DOM 参考手册：backgroundPosition 属性

CSS background-repeat 属性

实例

```
body
  {
  background-image: url(stars.gif);
  background-repeat: repeat-y;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 background-repeat 属性。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

background-repeat 属性设置是否及如何重复背景图像。

默认地，背景图像在水平和垂直方向上重复。

详细说明

background-repeat 属性定义了图像的平铺模式。

从原图像开始重复，原图像由 background-image 定义，并根据 background-position 的值放置。

默认值：	repeat
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.backgroundRepeat="repeat-y"

提示和注释

提示：背景图像的位置是根据 background-position 属性设置的。如果未规定 background-position 属性，图像会被放置在元素的左上角。

可能的值

值	描述
repeat	默认。背景图像将在垂直方向和水平方向重复。
repeat-x	背景图像将在水平方向重复。
repeat-y	背景图像将在垂直方向重复。
no-repeat	背景图像将仅显示一次。
inherit	规定应该从父元素继承 background-repeat 属性的设置。

TIY 实例

如何重复背景图像

本例演示如何重复背景图像。

```
<html>
<head>

<style type="text/css">
body
{
background-image:
url(/i/eg_bg_03.gif);
background-repeat: repeat
}
</style>

</head>

<body>
</body>
</html>
```

如何在垂直方向重复背景图像

本例演示如何垂直地重复背景图像。

```
<html>
<head>

<style type="text/css">
body
{
background-image:
url(/i/eg_bg_03.gif);
background-repeat: repeat-y
}
</style>

</head>

<body>
</body>
</html>
```

如何在水平方向重复背景图像

本例演示如何水平地重复背景图像。

```
<html>
<head>

<style type="text/css">
body
{
background-image:
url(/i/eg_bg_03.gif);
background-repeat: repeat-x
}
</style>

</head>

<body>
</body>
</html>
```

如何仅显示一次背景图像

本例演示如何仅显示一次背景图像。

```
<html>
<head>

<style type="text/css">
body
{
background-image: url('/i/eg_bg_03.gif');
background-repeat: no-repeat
}
</style>

</head>

<body>
</body>
</html>
```

相关页面

CSS 教程：CSS 背景

HTML DOM 参考手册：backgroundRepeat 属性

CSS3 background-clip 属性

实例

规定背景的绘制区域：

```
div
{
background-color:yellow;
background-clip:content-box;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

IE9+、Firefox、Opera、Chrome 以及 Safari 支持 background-clip 属性。

注释：Internet Explorer 8 以及更早的版本不支持 background-clip 属性。

定义和用法

background-clip 属性规定背景的绘制区域。

默认值：	**border-box**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.backgroundClip="content-box"

语法

```
background-clip: border-box|padding-box|content-box;
```

值	描述	测试
border-box	背景被裁剪到边框盒。	测试
padding-box	背景被裁剪到内边距框。	测试
content-box	背景被裁剪到内容框。	测试

相关页面

CSS3 教程：CSS3 背景

CSS3 background-origin 属性

实例

相对于内容框来定位背景图像：

```
div
{
background-image:url('smiley.gif');
background-repeat:no-repeat;
background-position:left;
background-origin:content-box;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

IE9+、Firefox 4+、Opera、Chrome 以及 Safari 5+ 支持 background-origin 属性。

定义和用法

background-origin 属性规定 background-position 属性相对于什么位置来定位。

注释：如果背景图像的 background-attachment 属性为 "fixed"，则该属性没有效果。

默认值：	**padding-box**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.backgroundOrigin="content-box"

语法

```
background-origin: padding-box|border-box|content-box;
```

值	描述	测试
padding-box	背景图像相对于内边距框来定位。	测试
border-box	背景图像相对于边框盒来定位。	测试
content-box	背景图像相对于内容框来定位。	测试

相关页面

CSS3 教程：CSS3 背景

CSS3 background-size 属性

实例

规定背景图像的尺寸：

```
div
{
background:url(img_flwr.gif);
background-size:80px 60px;
background-repeat:no-repeat;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

IE9+、Firefox 4+、Opera、Chrome 以及 Safari 5+ 支持 background-size 属性。

定义和用法

background-size 属性规定背景图像的尺寸。

默认值：	auto
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.backgroundSize="60px 80px"

语法

```
background-size: _length_|_percentage_|cover|contain;
```

值	描述	测试
length		

设置背景图像的高度和宽度。

第一个值设置宽度，第二个值设置高度。

如果只设置一个值，则第二个值会被设置为 "auto"。

| 测试 | | *percentage* |

以父元素的百分比来设置背景图像的宽度和高度。

第一个值设置宽度，第二个值设置高度。

如果只设置一个值，则第二个值会被设置为 "auto"。

| 测试 | | cover |

把背景图像扩展至足够大，以使背景图像完全覆盖背景区域。

背景图像的某些部分也许无法显示在背景定位区域中。

| 测试 | | contain | 把图像图像扩展至最大尺寸，以使其宽度和高度完全适应内容区域。 | 测试 |

亲自试一试 - 实例

拉伸背景图像

拉伸背景图像来完全覆盖内容区域。

```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
background:url(/i/bg_flower.gif);
background-size:35% 100%;
-moz-background-size:35% 100%; /* 老版本的 Firefox */
background-repeat:no-repeat;
}
</style>
</head>
<body>

<div>
<p>这是一个段落。</p>
<p>这是一个段落。</p>
<p>这是一个段落。</p>
<p>这是一个段落。</p>
<p>这是一个段落。</p>
<p>这是一个段落。</p>
<p>这是一个段落。</p>
<p>这是一个段落。</p>
<p>这是一个段落。</p>
</div>

</body>
</html>
```

拉伸背景图像，对背景图像水平复制四次

对背景图像进行拉伸，以使背景图像恰好水平复制四次。

```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
background:url(/i/bg_flower.gif);
background-size:25%;
border:2px solid #92b901;
}
</style>
</head>
<body>

<div>
<p>这是一个段落。这是一个段落。这是一个段落。这是一个段落。这是一个段落。</p>
<p>这是一个段落。这是一个段落。这是一个段落。这是一个段落。这是一个段落。</p>
<p>这是一个段落。这是一个段落。这是一个段落。这是一个段落。这是一个段落。</p>
<p>这是一个段落。这是一个段落。这是一个段落。这是一个段落。这是一个段落。</p>
<p>这是一个段落。这是一个段落。这是一个段落。这是一个段落。这是一个段落。</p>
<p>这是一个段落。这是一个段落。这是一个段落。这是一个段落。这是一个段落。</p>
</div>

</body>
</html>
```

相关页面

CSS3 教程：CSS3 背景

CSS 边框属性（Border 和 Outline）

属性	描述	CSS
border	在一个声明中设置所有的边框属性。	1
border-bottom	在一个声明中设置所有的下边框属性。	1
border-bottom-color	设置下边框的颜色。	2
border-bottom-style	设置下边框的样式。	2
border-bottom-width	设置下边框的宽度。	1
border-color	设置四条边框的颜色。	1
border-left	在一个声明中设置所有的左边框属性。	1
border-left-color	设置左边框的颜色。	2
border-left-style	设置左边框的样式。	2
border-left-width	设置左边框的宽度。	1
border-right	在一个声明中设置所有的右边框属性。	1
border-right-color	设置右边框的颜色。	2
border-right-style	设置右边框的样式。	2
border-right-width	设置右边框的宽度。	1
border-style	设置四条边框的样式。	1
border-top	在一个声明中设置所有的上边框属性。	1
border-top-color	设置上边框的颜色。	2
border-top-style	设置上边框的样式。	2
border-top-width	设置上边框的宽度。	1
border-width	设置四条边框的宽度。	1
outline	在一个声明中设置所有的轮廓属性。	2
outline-color	设置轮廓的颜色。	2
outline-style	设置轮廓的样式。	2
outline-width	设置轮廓的宽度。	2
border-bottom-left-radius	定义边框左下角的形状。	3
border-bottom-right-radius	定义边框右下角的形状。	3

border-image	简写属性，设置所有 border-image-* 属性。	3
border-image-outset	规定边框图像区域超出边框的量。	3
border-image-repeat	图像边框是否应平铺(repeated)、铺满(rounded)或拉伸(stretched)。	3
border-image-slice	规定图像边框的向内偏移。	3
border-image-source	规定用作边框的图片。	3
border-image-width	规定图片边框的宽度。	3
border-radius	简写属性，设置所有四个 border-*-radius 属性。	3
border-top-left-radius	定义边框左上角的形状。	3
border-top-right-radius	定义边框右下角的形状。	3
box-decoration-break	3	
box-shadow	向方框添加一个或多个阴影。	3

CSS border 属性

实例

设置 4 个边框的样式：

```
p
  {
  border:5px solid red;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border 属性。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border 简写属性在一个声明设置所有的边框属性。

可以按顺序设置如下属性：

- border-width
- border-style
- border-color

如果不设置其中的某个值，也不会出问题，比如 border:solid #ff0000; 也是允许的。

默认值：	***not specified***
继承性：	no
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.border="3px solid blue"
---	---

可能的值

值	描述
border-width	规定边框的宽度。参阅：border-width 中可能的值。
border-style	规定边框的样式。参阅：border-style 中可能的值。
border-color	规定边框的颜色。参阅：border-color 中可能的值。
inherit	规定应该从父元素继承 border 属性的设置。

TIY 实例

所有边框属性在一个声明之中

本例演示用简写属性来将所有四个边框属性设置于同一声明中。

```
<html>
<head>
<style type="text/css">
p
{
border: medium double rgb(250,0,255)
}
</style>
</head>

<body>
<p>Some text</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 边框

HTML DOM 参考手册：border 属性

CSS border-bottom 属性

实例

设置下边框的样式：

```
p
  {
  border-style:solid;
  border-bottom:thick dotted #ff0000;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-bottom 属性。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-bottom 简写属性把下边框的所有属性设置到一个声明中。

可以按顺序设置如下属性：

- border-bottom-width
- border-bottom-style
- border-bottom-color

如果不设置其中的某个值，也不会出问题，比如 border-bottom:solid #ff0000; 也是允许的。

默认值：	***not specified***
继承性：	no
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.borderBottom="3px solid blue"
---	---

可能的值

值	描述
border-bottom-width	规定下边框的宽度。参阅：border-bottom-width 中可能的值。
border-bottom-style	规定下边框的样式。参阅：border-bottom-style 中可能的值。
border-bottom-color	规定下边框的颜色。参阅：border-bottom-color 中可能的值。
inherit	规定应该从父元素继承 border-bottom 属性的设置。

TIY 实例

所有下边框属性在一个声明中

本例演示用简写属性来将所有下边框属性设置在同一声明中。

```
<html>
<head>
<style type="text/css">
p
{
border-style:solid;
border-bottom:thick dotted #ff0000;
}
</style>
</head>

<body>
<p>This is some text in a paragraph.</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 边框

HTML DOM 参考手册：borderBottom 属性

CSS border-bottom-color 属性

实例

设置下边框的颜色：

```
p
  {
  border-style:solid;
  border-bottom-color:#ff0000;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-bottom-color 属性。

注释：Internet Explorer 6（以及更早的版本）不支持属性值 "transparent"。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-bottom-color 设置元素的下边框的颜色。

只能定义纯色，而且只有当边框的样式是一个非 none 或 hidden 的值时边框才可能出现。

注释：请始终把 border-style 属性声明到 border-color 属性之前。元素必须在您改变其颜色之前获得边框。

默认值：	***not specified***
继承性：	no
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.borderBottomColor="blue"
---	---

可能的值

值	描述
color_name	规定颜色值为颜色名称的边框颜色（比如 red）。
hex_number	规定颜色值为十六进制值的边框颜色（比如 #ff0000）。
rgb_number	规定颜色值为 rgb 代码的边框颜色（比如 rgb(255,0,0)）。
transparent	默认值。边框颜色为透明。
inherit	规定应该从父元素继承边框颜色。

TIY 实例

设置下边框的颜色

本例演示如何设置下边框的颜色。

```
<html>
<head>
<style type="text/css">
p
{
border-style:solid;
border-bottom-color:#ff0000;
}
</style>
</head>

<body>
<p>This is some text in a paragraph.</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 边框

CSS 参考手册：border-bottom 属性

HTML DOM 参考手册：borderBottomColor 属性

CSS border-bottom-style 属性

实例

设置下边框的样式：

```
p
  {
  border-style:solid;
  border-bottom-style:dotted;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-bottom-style 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit" 或 "hidden"。

定义和用法

border-bottom-style 设置元素下边框的样式。

只有当这个值不是 none 时边框才可能出现。

在 CSS1 中，HTML 用户代理只需支持 solid 和 none。

默认值：	*not specified*
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderBottomStyle="dotted"

可能的值

值	描述
none	定义无边框。
hidden	与 "none" 相同。不过应用于表时除外，对于表，hidden 用于解决边框冲突。
dotted	定义点状边框。在大多数浏览器中呈现为实线。
dashed	定义虚线。在大多数浏览器中呈现为实线。
solid	定义实线。
double	定义双线。双线的宽度等于 border-width 的值。
groove	定义 3D 凹槽边框。其效果取决于 border-color 的值。
ridge	定义 3D 垄状边框。其效果取决于 border-color 的值。
inset	定义 3D inset 边框。其效果取决于 border-color 的值。
outset	定义 3D outset 边框。其效果取决于 border-color 的值。
inherit	规定应该从父元素继承边框样式。

TIY 实例

设置下边框的样式

本例演示如何设置下边框的样式。

```
<html>
<head>
<style type="text/css">
p {border-style:solid}
p.none {border-bottom-style:none}
p.dotted {border-bottom-style:dotted}
p.dashed {border-bottom-style:dashed}
p.solid {border-bottom-style:solid}
p.double {border-bottom-style:double}
p.groove {border-bottom-style:groove}
p.ridge {border-bottom-style:ridge}
p.inset {border-bottom-style:inset}
p.outset {border-bottom-style:outset}
</style>
</head>

<body>
<p class="none">No bottom border.</p>
<p class="dotted">A dotted bottom border.</p>
<p class="dashed">A dashed bottom border.</p>
<p class="solid">A solid bottom border.</p>
<p class="double">A double bottom border.</p>
<p class="groove">A groove bottom border.</p>
<p class="ridge">A ridge bottom border.</p>
<p class="inset">An inset bottom border.</p>
<p class="outset">An outset bottom border.</p>
</body>
</html>
```

相关页面

CSS 教程：CSS 边框

CSS 参考手册：border-bottom 属性

HTML DOM 参考手册：borderBottomStyle 属性

CSS border-bottom-width 属性

实例

设置下边框的宽度：

```
p
  {
  border-style:solid;
  border-bottom-width:15px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-bottom-width 属性。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-bottom-width 属性设置元素的下边框的宽度。

只有当边框样式不是 none 时才起作用。如果边框样式是 none，边框宽度实际上会重置为 0。不允许指定负长度值。

注释：请始终在 border-bottom-width 属性之前声明 border-style 属性。元素只有在获得边框之后，才能改变其边框的宽度。

默认值：	medium
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderBottomWidth="thick"

可能的值

值	描述
thin	定义细的下边框。
medium	默认值。定义中等的下边框。
thick	定义粗的下边框。
length	允许您自定义下边框的宽度。
inherit	规定应该从父元素继承边框宽度。

TIY 实例

设置下边框的宽度

本例演示如何设置下边框的宽度。

```
<html>
<head>
<style type="text/css">
p.one
{
border-style: solid;
border-bottom-width: 15px
}
p.two
{
border-style: solid;
border-bottom-width: thin
}
</style>
</head>
<body>

<p class="one"><b>注释：</b>"border-bottom-width" 属性如果单独使用的话是不会起作用的。请首先使用 '
<p class="two">Some text. Some more text.</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 边框

CSS 参考手册：border-bottom 属性

HTML DOM 参考手册：borderBottomWidth 属性

CSS border-color 属性

实例

设置 4 个边框的颜色：

```
p
  {
  border-style:solid;
  border-color:#ff0000 #0000ff;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-color 属性。

注释：Internet Explorer 6（以及更早的版本）不支持属性值 "transparent"。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-color 属性设置四条边框的颜色。此属性可设置 1 到 4 种颜色。

border-color 属性是一个简写属性，可设置一个元素的所有边框中可见部分的颜色，或者为 4 个边分别设置不同的颜色。请看下面的例子：

例子 1

```
border-color:red green blue pink;
```

- 上边框是红色
- 右边框是绿色
- 下边框是蓝色
- 左边框是粉色

例子 2

```
border-color:red green blue;
```

- 上边框是红色
- 右边框和左边框是绿色
- 下边框是蓝色

例子 3

```
border-color:dotted red green;
```

- 上边框和下边框是红色
- 右边框和左边框是绿色

例子 4

```
border-color:red;
```

- 所有 4 个边框都是红色

要记住，边框的样式不能为 none 或 hidden，否则边框不会出现。

注释：请始终把 border-style 属性声明到 border-color 属性之前。元素必须在您改变其颜色之前获得边框。

默认值：	***not specified***
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderColor="#FF0000 blue"

可能的值

值	描述
color_name	规定颜色值为颜色名称的边框颜色（比如 red）。
hex_number	规定颜色值为十六进制值的边框颜色（比如 #ff0000）。
rgb_number	规定颜色值为 rgb 代码的边框颜色（比如 rgb(255,0,0)）。
transparent	默认值。边框颜色为透明。
inherit	规定应该从父元素继承边框颜色。

TIY 实例

设置四个边框的颜色

本例演示如何设置四个边框的颜色。可以设置一到四种颜色。

```
<html>
<head>

<style type="text/css">
p.one
{
border-style: solid;
border-color: #0000ff
}
p.two
{
border-style: solid;
border-color: #ff0000 #0000ff
}
p.three
{
border-style: solid;
border-color: #ff0000 #00ff00 #0000ff
}
p.four
{
border-style: solid;
border-color: #ff0000 #00ff00 #0000ff rgb(250,0,255)
}
</style>

</head>

<body>

<p class="one">One-colored border!</p>

<p class="two">Two-colored border!</p>

<p class="three">Three-colored border!</p>

<p class="four">Four-colored border!</p>

<p><b>注释：</b>"border-width" 属性如果单独使用的话是不会起作用的。请首先使用 "border-style" 属性来

</body>
</html>
```

相关页面

CSS 教程：CSS 边框

HTML DOM 参考手册：borderColor 属性

CSS border-left 属性

实例

设置左边框的样式：

```
p
  {
  border-style:solid;
  border-left:thick double #ff0000;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-left 属性。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-left 简写属性把左边框的所有属性设置到一个声明中。

可以按顺序设置如下属性：

- border-left-width
- border-left-style
- border-left-color

如果不设置其中的某个值，也不会出问题，比如 border-left:solid #ff0000; 也是允许的。

默认值：	*not specified*
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderLeft="3px solid blue"

可能的值

值	描述
border-left-width	规定左边框的宽度。参阅：border-left-width 中可能的值。
border-left-style	规定左边框的样式。参阅：border-left-style 中可能的值。
border-left-color	规定左边框的颜色。参阅：border-left-color 中可能的值。
inherit	规定应该从父元素继承 border-left 属性的设置。

TIY 实例

所有左边框属性在一个声明之中

所有左边框属性在一个声明之中

```
<html>
<head>
<style type="text/css">
p
{
border-style:solid;
border-left:thick double #ff0000;
}
</style>
</head>

<body>
<p>This is some text in a paragraph.</p>
</body>
</html>
```

相关页面

CSS 教程：CSS 边框

HTML DOM 参考手册：borderLeft 属性

CSS border-left-color 属性

实例

设置左边框的颜色：

```
p
  {
  border-style:solid;
  border-left-color:#ff0000;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-left-color 属性。

注释：Internet Explorer 6（以及更早的版本）不支持属性值 "transparent"。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-left-color 设置元素的左边框的颜色。

只能定义纯色，而且只有当边框的样式是一个非 none 或 hidden 的值时边框才可能出现。

注释：请始终把 border-style 属性声明到 border-color 属性之前。元素必须在您改变其颜色之前获得边框。

默认值：	*not specified*
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderLeftColor="blue"

可能的值

值	描述
color_name	规定颜色值为颜色名称的边框颜色（比如 red）。
hex_number	规定颜色值为十六进制值的边框颜色（比如 #ff0000）。
rgb_number	规定颜色值为 rgb 代码的边框颜色（比如 rgb(255,0,0)）。
transparent	默认值。边框颜色为透明。
inherit	规定应该从父元素继承边框颜色。

TIY 实例

设置左边框的颜色

本例演示如何设置左边框的颜色。

```
<html>
<head>
<style type="text/css">
p
{
border-style: solid;
border-left-color: #ff0000
}
</style>
</head>

<body>
<p>Some text.</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 边框

CSS 参考手册：border-left 属性

HTML DOM 参考手册：borderLeftColor 属性

CSS border-left-style 属性

实例

设置左边框的样式：

```
p
  {
  border-style:solid;
  border-left-style:dotted;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-left-style 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit" 或 "hidden"。

定义和用法

border-left-style 设置元素左边框的样式。

只有当这个值不是 none 时边框才可能出现。

在 CSS1 中，HTML 用户代理只需支持 solid 和 none。

默认值：	***not specified***
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderLeftStyle="dotted"

可能的值

值	描述
none	定义无边框。
hidden	与 "none" 相同。不过应用于表时除外，对于表，hidden 用于解决边框冲突。
dotted	定义点状边框。在大多数浏览器中呈现为实线。
dashed	定义虚线。在大多数浏览器中呈现为实线。
solid	定义实线。
double	定义双线。双线的宽度等于 border-width 的值。
groove	定义 3D 凹槽边框。其效果取决于 border-color 的值。
ridge	定义 3D 垄状边框。其效果取决于 border-color 的值。
inset	定义 3D inset 边框。其效果取决于 border-color 的值。
outset	定义 3D outset 边框。其效果取决于 border-color 的值。
inherit	规定应该从父元素继承边框样式。

TIY 实例

设置左边框的样式

本例演示如何设置左边框的样式。

```
<html>
<head>
<style type="text/css">
p
{
border-style:solid;
}
p.none {border-left-style:none}
p.dotted {border-left-style:dotted}
p.dashed {border-left-style:dashed}
p.solid {border-left-style:solid}
p.double {border-left-style:double}
p.groove {border-left-style:groove}
p.ridge {border-left-style:ridge}
p.inset {border-left-style:inset}
p.outset {border-left-style:outset}
</style>
</head>

<body>
<p class="none">No left border.</p>
<p class="dotted">A dotted left border.</p>
<p class="dashed">A dashed left border.</p>
<p class="solid">A solid left border.</p>
<p class="double">A double left border.</p>
<p class="groove">A groove left border.</p>
<p class="ridge">A ridge left border.</p>
<p class="inset">An inset left border.</p>
<p class="outset">An outset left border.</p>
</body>
</html>
```

相关页面

CSS 教程：CSS 边框

CSS 参考手册：border-left 属性

HTML DOM 参考手册：borderLeftStyle 属性

CSS border-left-width 属性

实例

设置左边框的宽度：

```
p
  {
  border-style:solid;
  border-left-width:15px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-left-width 属性。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-left-width 属性设置元素的左边框的宽度。

只有当边框样式不是 none 时才起作用。如果边框样式是 none，边框宽度实际上会重置为 0。不允许指定负长度值。

注释：请始终在 border-left-width 属性之前声明 border-style 属性。元素只有在获得边框之后，才能改变其边框的宽度。

默认值：	medium
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderLeftWidth="thick"

可能的值

值	描述
thin	定义细的左边框。
medium	默认值。定义中等的左边框。
thick	定义粗的左边框。
length	允许您自定义左边框的宽度。
inherit	规定应该从父元素继承边框宽度。

TIY 实例

设置左边框的宽度

本例演示如何设置左边框的宽度。

```
<html>
<head>
<style type="text/css">
p.one
{
border-style: solid;
border-left-width: 15px
}
p.two
{
border-style: solid;
border-left-width: thin
}
</style>
</head>
<body>

<p class="one"><b>注释：</b>"border-left-width" 属性如果单独使用的话是不会起作用的。请首先使用 "bo
<p class="two">Some text. Some more text.</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 边框

CSS 参考手册：border-left 属性

HTML DOM 参考手册：borderLeftWidth 属性

CSS border-right 属性

实例

设置右边框的样式：

```
p
  {
  border-style:solid;
  border-right:thick double #ff0000;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-right 属性。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-right 简写属性把右边框的所有属性设置到一个声明中。

可以按顺序设置如下属性：

- border-right-width
- border-right-style
- border-right-color

如果不设置其中的某个值，也不会出问题，比如 border-right:solid #ff0000; 也是允许的。

默认值：	*not specified*
继承性：	no
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.borderRight="3px solid blue"
---	---

可能的值

值	描述
border-right-width	规定右边框的宽度。参阅：border-right-width 中可能的值。
border-right-style	规定右边框的样式。参阅：border-right-style 中可能的值。
border-right-color	规定右边框的颜色。参阅：border-right-color 中可能的值。
inherit	规定应该从父元素继承 border-right 属性的设置。

TIY 实例

所有右边框属性在一个声明之中

本例演示一个简写属性，用于把所有右边框属性设置在一条声明中。

```
<html>
<head>
<style type="text/css">
p
{
border-style:solid;
border-right:thick double #ff0000;
}
</style>
</head>

<body>
<p>This is some text in a paragraph.</p>
</body>
</html>
```

相关页面

CSS 教程：CSS 边框

HTML DOM 参考手册：borderRight 属性

CSS border-right-color 属性

实例

设置右边框的颜色：

```
p
  {
  border-style:solid;
  border-right-color:#ff0000;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-right-color 属性。

注释：Internet Explorer 6（以及更早的版本）不支持属性值 "transparent"。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-right-color 设置元素的右边框的颜色。

只能定义纯色，而且只有当边框的样式是一个非 none 或 hidden 的值时边框才可能出现。

注释：请始终把 border-style 属性声明到 border-color 属性之前。元素必须在您改变其颜色之前获得边框。

默认值：	*not specified*
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderRightColor="blue"

可能的值

值	描述
color_name	规定颜色值为颜色名称的边框颜色（比如 red）。
hex_number	规定颜色值为十六进制值的边框颜色（比如 #ff0000）。
rgb_number	规定颜色值为 rgb 代码的边框颜色（比如 rgb(255,0,0)）。
transparent	默认值。边框颜色为透明。
inherit	规定应该从父元素继承边框颜色。

TIY 实例

设置右边框的颜色

本例演示如何设置右边框的颜色。

```
<html>
<head>
<style type="text/css">
p
{
border-style: solid;
border-right-color: #ff0000
}
</style>
</head>

<body>
<p>Some text.</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 边框

CSS 参考手册：border-right 属性

HTML DOM 参考手册：borderRightColor 属性

CSS border-right-style 属性

实例

设置右边框的样式：

```
p
  {
  border-style:solid;
  border-right-style:dotted;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-right-style 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit" 或 "hidden"。

定义和用法

border-right-style 设置元素右边框的样式。

只有当这个值不是 none 时边框才可能出现。

在 CSS1 中，HTML 用户代理只需支持 solid 和 none。

默认值：	***not specified***
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderRightStyle="dotted"

可能的值

值	描述
none	定义无边框。
hidden	与 "none" 相同。不过应用于表时除外，对于表，hidden 用于解决边框冲突。
dotted	定义点状边框。在大多数浏览器中呈现为实线。
dashed	定义虚线。在大多数浏览器中呈现为实线。
solid	定义实线。
double	定义双线。双线的宽度等于 border-width 的值。
groove	定义 3D 凹槽边框。其效果取决于 border-color 的值。
ridge	定义 3D 垄状边框。其效果取决于 border-color 的值。
inset	定义 3D inset 边框。其效果取决于 border-color 的值。
outset	定义 3D outset 边框。其效果取决于 border-color 的值。
inherit	规定应该从父元素继承边框样式。

TIY 实例

设置右边框的样式

本例演示如何设置右边框的样式。

```
<!DOCTYPE html>

<html>
<head>
<style type="text/css">
p.dotted {border-right-style: dotted}
p.dashed {border-right-style: dashed}
p.solid {border-right-style: solid}
p.double {border-right-style: double}
p.groove {border-right-style: groove}
p.ridge {border-right-style: ridge}
p.inset {border-right-style: inset}
p.outset {border-right-style: outset}
</style>
</head>

<body>
<p class="dotted">A dotted border</p>

<p class="dashed">A dashed border</p>

<p class="solid">A solid border</p>

<p class="double">A double border</p>

<p class="groove">A groove border</p>

<p class="ridge">A ridge border</p>

<p class="inset">An inset border</p>

<p class="outset">An outset border</p>
</body>
</html>
```

相关页面

CSS 教程：CSS 边框

CSS 参考手册：border-right 属性

HTML DOM 参考手册：borderRightStyle 属性

CSS border-right-width 属性

实例

设置右边框的宽度：

```
p
  {
  border-style:solid;
  border-right-width:15px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-right-width 属性。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-right-width 属性设置元素的右边框的宽度。

只有当边框样式不是 none 时才起作用。如果边框样式是 none，边框宽度实际上会重置为 0。不允许指定负长度值。

注释：请始终在 border-right-width 属性之前声明 border-style 属性。元素只有在获得边框之后，才能改变其边框的宽度。

默认值：	medium
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderRightWidth="thick"

可能的值

值	描述
thin	定义细的右边框。
medium	默认值。定义中等的右边框。
thick	定义粗的右边框。
length	允许您自定义右边框的宽度。
inherit	规定应该从父元素继承边框宽度。

TIY 实例

设置右边框的宽度

本例演示如何设置右边框的宽度。

```
<html>
<head>
<style type="text/css">
p.one
{
border-style: solid;
border-right-width: 15px
}
p.two
{
border-style: solid;
border-right-width: thin
}
</style>
</head>
<body>

<p class="one"><b>注释：</b>"border-right-width" 属性如果单独使用的话是不会起作用的。请首先使用 "b
<p class="two">Some text. Some more text.</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 边框

CSS 参考手册：border-right 属性

HTML DOM 参考手册：borderRightWidth 属性

CSS border-style 属性

实例

设置 4 个边框的样式：

```
p
  {
  border-style:solid;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-style 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit" 或 "hidden"。

定义和用法

border-style 属性用于设置元素所有边框的样式，或者单独地为各边设置边框样式。

只有当这个值不是 none 时边框才可能出现。

例子 1

```
border-style:dotted solid double dashed;
```

- 上边框是点状
- 右边框是实线
- 下边框是双线
- 左边框是虚线

例子 2

```
border-style:dotted solid double;
```

- 上边框是点状

- 右边框和左边框是实线
- 下边框是双线

例子 3

```
border-style:dotted solid;
```

- 上边框和下边框是点状
- 右边框和左边框是实线

例子 4

```
border-style:dotted;
```

- 所有 4 个边框都是点状

默认值：	***not specified***
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderStyle="dotted double"

可能的值

值	描述
none	定义无边框。
hidden	与 "none" 相同。不过应用于表时除外，对于表，hidden 用于解决边框冲突。
dotted	定义点状边框。在大多数浏览器中呈现为实线。
dashed	定义虚线。在大多数浏览器中呈现为实线。
solid	定义实线。
double	定义双线。双线的宽度等于 border-width 的值。
groove	定义 3D 凹槽边框。其效果取决于 border-color 的值。
ridge	定义 3D 垄状边框。其效果取决于 border-color 的值。
inset	定义 3D inset 边框。其效果取决于 border-color 的值。
outset	定义 3D outset 边框。其效果取决于 border-color 的值。
inherit	规定应该从父元素继承边框样式。

描述

最不可预测的边框样式是 double。它定义为两条线的宽度再加上这两条线之间的空间等于 border-width 值。不过，CSS 规范并没有说其中一条线是否比另一条粗或者两条线是否应该是一样的粗，也没有指出线之间的空间是否应当比线粗。所有这些都有用户代理决定，创作人员对这个决定没有任何影响。

TIY 实例

设置四边框样式

本例演示如何设置四边框样式。

```
<html>
<head>
<style type="text/css">
p.dotted {border-style: dotted}
p.dashed {border-style: dashed}
p.solid {border-style: solid}
p.double {border-style: double}
p.groove {border-style: groove}
p.ridge {border-style: ridge}
p.inset {border-style: inset}
p.outset {border-style: outset}
</style>
</head>

<body>
<p class="dotted">A dotted border</p>

<p class="dashed">A dashed border</p>

<p class="solid">A solid border</p>

<p class="double">A double border</p>

<p class="groove">A groove border</p>

<p class="ridge">A ridge border</p>

<p class="inset">An inset border</p>

<p class="outset">An outset border</p>
</body>

</html>
```

设置每一边的不同边框

本例演示如何在元素的各边设置不同的边框。

```
<html>
<head>
<style type="text/css">
p.soliddouble {border-style: solid double}
p.doublesolid {border-style: double solid}
p.groovedouble {border-style: groove double}
p.three {border-style: solid double groove}
</style>
</head>

<body>
<p class="soliddouble">Some text</p>

<p class="doublesolid">Some text</p>

<p class="groovedouble">Some text</p>

<p class="three">Some text</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 边框

HTML DOM 参考手册：borderStyle 属性

CSS border-top 属性

实例

设置上边框的样式：

```
p
  {
  border-style:solid;
  border-top:thick double #ff0000;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-top 属性。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-top 简写属性把上边框的所有属性设置到一个声明中。

可以按顺序设置如下属性：

- border-top-width
- border-top-style
- border-top-color

如果不设置其中的某个值，也不会出问题，比如 border-top:solid #ff0000; 也是允许的。

默认值：	***not specified***
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderTop="3px solid blue"

可能的值

值	描述
border-top-width	规定上边框的宽度。参阅：border-top-width 中可能的值。
border-top-style	规定上边框的样式。参阅：border-top-style 中可能的值。
border-top-color	规定上边框的颜色。参阅：border-top-color 中可能的值。
inherit	规定应该从父元素继承 border-top 属性的设置。

TIY 实例

所有上边框属性在一个声明之中

本例演示用简写属性来将所有上边框属性设置于同一声明之中。

```
<html>
<head>
<style type="text/css">
p
{
border-style:solid;
border-top:thick double #ff0000;
}
</style>
</head>

<body>
<p>This is some text in a paragraph.</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 边框

HTML DOM 参考手册：borderTop 属性

CSS border-top-color 属性

实例

设置上边框的颜色：

```
p
  {
  border-style:solid;
  border-top-color:#ff0000;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-top-color 属性。

注释：Internet Explorer 6（以及更早的版本）不支持属性值 "transparent"。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-top-color 设置元素的上边框的颜色。

只能定义纯色，而且只有当边框的样式是一个非 none 或 hidden 的值时边框才可能出现。

注释：请始终把 border-style 属性声明到 border-color 属性之前。元素必须在您改变其颜色之前获得边框。

默认值：	*not specified*
继承性：	no
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.borderTopColor="blue"
---	---

可能的值

值	描述
color_name	规定颜色值为颜色名称的边框颜色（比如 red）。
hex_number	规定颜色值为十六进制值的边框颜色（比如 #ff0000）。
rgb_number	规定颜色值为 rgb 代码的边框颜色（比如 rgb(255,0,0)）。
transparent	默认值。边框颜色为透明。
inherit	规定应该从父元素继承边框颜色。

TIY 实例

设置上边框的颜色

本例演示如何设置上边框的颜色。

```
<html>
<head>
<style type="text/css">
p
{
border-style: solid;
border-top-color: #ff0000
}
</style>
</head>

<body>
<p>Some text.</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 边框

CSS 参考手册：border-top 属性

HTML DOM 参考手册：borderTopColor 属性

CSS border-top-style 属性

实例

设置上边框的样式：

```
p
  {
  border-style:solid;
  border-top-style:dotted;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-top-style 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit" 或 "hidden"。

定义和用法

border-top-style 设置元素上边框的样式。

只有当这个值不是 none 时边框才可能出现。

在 CSS1 中，HTML 用户代理只需支持 solid 和 none。

默认值：	***not specified***
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderTopStyle="dotted"

可能的值

值	描述
none	定义无边框。
hidden	与 "none" 相同。不过应用于表时除外，对于表，hidden 用于解决边框冲突。
dotted	定义点状边框。在大多数浏览器中呈现为实线。
dashed	定义虚线。在大多数浏览器中呈现为实线。
solid	定义实线。
double	定义双线。双线的宽度等于 border-width 的值。
groove	定义 3D 凹槽边框。其效果取决于 border-color 的值。
ridge	定义 3D 垄状边框。其效果取决于 border-color 的值。
inset	定义 3D inset 边框。其效果取决于 border-color 的值。
outset	定义 3D outset 边框。其效果取决于 border-color 的值。
inherit	规定应该从父元素继承边框样式。

TIY 实例

设置上边框的样式

本例演示如何设置上边框的样式。

```
<html>
<head>
<style type="text/css">
p
{
border-style:solid;
}
p.none {border-top-style:none}
p.dotted {border-top-style:dotted}
p.dashed {border-top-style:dashed}
p.solid {border-top-style:solid}
p.double {border-top-style:double}
p.groove {border-top-style:groove}
p.ridge {border-top-style:ridge}
p.inset {border-top-style:inset}
p.outset {border-top-style:outset}
</style>
</head>

<body>
<p class="none">No top border.</p>
<p class="dotted">A dotted top border.</p>
<p class="dashed">A dashed top border.</p>
<p class="solid">A solid top border.</p>
<p class="double">A double top border.</p>
<p class="groove">A groove top border.</p>
<p class="ridge">A ridge top border.</p>
<p class="inset">An inset top border.</p>
<p class="outset">An outset top border.</p>
</body>
</html>
```

相关页面

CSS 教程：CSS 边框

CSS 参考手册：border-top 属性

HTML DOM 参考手册：borderTopStyle 属性

CSS border-top-width 属性

实例

设置上边框的宽度：

```
p
  {
  border-style:solid;
  border-top-width:15px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-top-width 属性。

注释：IE7 以及更早版本的浏览器不支持值 "inherit"。IE8 需要 !DOCTYPE。IE9 支持 "inherit"。

定义和用法

border-top-width 属性设置元素的上边框的宽度。

只有当边框样式不是 none 时才起作用。如果边框样式是 none，边框宽度实际上会重置为 0。不允许指定负长度值。

注释：请始终在 border-top-width 属性之前声明 border-style 属性。元素只有在获得边框之后，才能改变其边框的宽度。

默认值：	medium
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.borderTopWidth="thick"

可能的值

值	描述
thin	定义细的上边框。
medium	默认值。定义中等的上边框。
thick	定义粗的上边框。
length	允许您自定义上边框的宽度。
inherit	规定应该从父元素继承边框宽度。

TIY 实例

设置上边框的宽度

本例演示如何设置上边框的宽度。

```
<html>
<head>
<style type="text/css">
p.one
{
border-style: solid;
border-top-width: 15px
}
p.two
{
border-style: solid;
border-top-width: thin
}
</style>
</head>

<body>
<p class="one"><b>注释：</b>"border-top-width" 属性如果单独使用的话是不会起作用的。请首先使用 "bor
<p class="two">Some text. Some more text.</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 边框

CSS 参考手册：border-top 属性

HTML DOM 参考手册：borderTopWidth 属性

CSS border-width 属性

实例

设置四个边框的宽度：

```
p
  {
  border-style:solid;
  border-width:15px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 border-width 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

border-width 简写属性为元素的所有边框设置宽度，或者单独地为各边边框设置宽度。

只有当边框样式不是 none 时才起作用。如果边框样式是 none，边框宽度实际上会重置为 0。不允许指定负长度值。

例子 1

```
border-width:thin medium thick 10px;
```

- 上边框是细边框
- 右边框是中等边框
- 下边框是粗边框
- 左边框是 10px 宽的边框

例子 2

```
border-width:thin medium thick;
```

- 上边框是 10px
- 右边框和左边框是中等边框
- 下边框是粗边框

例子 3

```
border-width:thin medium;
```

- 上边框和下边框是细边框
- 右边框和左边框是中等边框

例子 4

```
border-width:thin;
```

- 所有 4 个边框都是细边框

默认值：	medium
继承性：	no
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.borderWidth="thin thick"
---	---

可能的值

值	描述
thin	定义细的边框。
medium	默认。定义中等的边框。
thick	定义粗的边框。
length	允许您自定义边框的宽度。
inherit	规定应该从父元素继承边框宽度。

TIY 实例

所有边框宽度属性在一个声明之中

本例演示用简写属性来将所有边框宽度属性设置于同一声明中。

```
<html>
<head>
<style type="text/css">
p.one
{
border-style: solid;
border-width: 5px
}
p.two
{
border-style: solid;
border-width: thick
}
p.three
{
border-style: solid;
border-width: 5px 10px
}
p.four
{
border-style: solid;
border-width: 5px 10px 1px
}
p.five
{
border-style: solid;
border-width: 5px 10px 1px medium
}
</style>
</head>

<body>
<p class="one">Some text</p>
<p class="two">Some text</p>
<p class="three">Some text</p>
<p class="four">Some text</p>
<p class="five">Some text</p>

<p><b>注释：</b>"border-width" 属性如果单独使用的话是不会起作用的。请首先使用 "border-style" 属性来
</body>

</html>
```

相关页面

CSS 教程：CSS 边框

HTML DOM 参考手册：borderWidth 属性

CSS outline 属性

实例

设置 4 个边框的样式：

```
p
  {
  outline:#00FF00 dotted thick;
  }
```

浏览器支持

所有浏览器都支持 outline 属性。

注释：如果规定了 !DOCTYPE，则 IE8 支持 outline 属性。

定义和用法

outline （轮廓）是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。

注释：轮廓线不会占据空间，也不一定是矩形。

outline 简写属性在一个声明中设置所有的轮廓属性。

可以按顺序设置如下属性：

- outline-color
- outline-style
- outline-width

如果不设置其中的某个值，也不会出问题，比如 outline:solid #ff0000; 也是允许的。

默认值：	invert none medium
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.outline="#0000FF dotted thin"

可能的值

值	描述
outline-color	规定边框的颜色。参阅：outline-color 中可能的值。
outline-style	规定边框的样式。参阅：outline-style 中可能的值。
outline-width	规定边框的宽度。参阅：outline-width 中可能的值。
inherit	规定应该从父元素继承 outline 属性的设置。

TIY 实例

在元素周围画线

本例演示使用 outline 属性在元素周围画一条线。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtm
<html>
<head>
<style type="text/css">
p
{
border:red solid thin;
outline:#00ff00 dotted thick;
}
</style>
</head>

<body>
<p><b>注释：</b>只有在规定了 !DOCTYPE 时，Internet Explorer 8 （以及更高版本） 才支持 outline 属性
</body>
</html>
```

相关页面

CSS 教程：CSS 轮廓

HTML DOM 参考手册：outline 属性

CSS outline-color 属性

实例

设置点状轮廓的颜色：

```
p
  {
  outline-style:dotted;
  outline-color:#00ff00;
  }
```

浏览器支持

所有浏览器都支持 outline-color 属性。

注释：如果规定了 !DOCTYPE，则 IE8 支持 outline 属性。

定义和用法

outline （轮廓）是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。outline 属性可设置元素周围的轮廓线。

注释：请始终在 outline-color 属性之前声明 outline-style 属性。元素只有获得轮廓以后才能改变其轮廓的颜色。

注释：轮廓线不会占据空间，也不一定是矩形。

outline-color 属性设置一个元素整个轮廓中可见部分的颜色。要记住，轮廓的样式不能是 none，否则轮廓不会出现。

默认值：	**invert**
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.outlineColor="#0000FF"

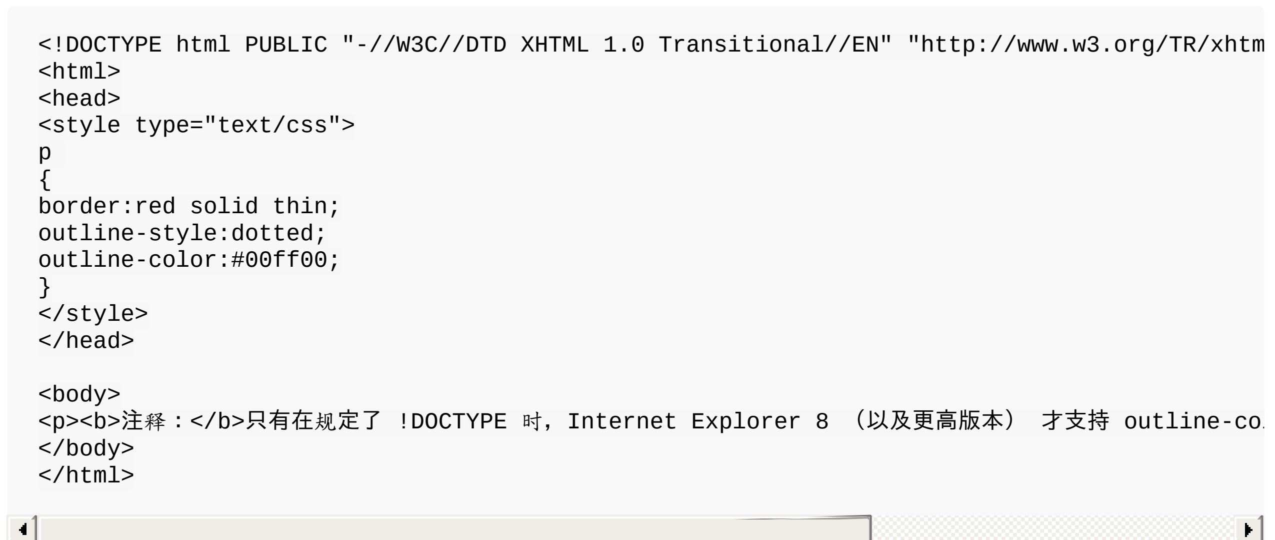
可能的值

值	描述
color_name	规定颜色值为颜色名称的轮廓颜色（比如 red）。
hex_number	规定颜色值为十六进制值的轮廓颜色（比如 #ff0000）。
rgb_number	规定颜色值为 rgb 代码的轮廓颜色（比如 rgb(255,0,0)）。
invert	默认。执行颜色反转（逆向的颜色）。可使轮廓在不同的背景颜色中都是可见。
inherit	规定应该从父元素继承轮廓颜色的设置。

TIY 实例

设置轮廓的颜色

本例演示如何设置轮廓的颜色。



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtm
<html>
<head>
<style type="text/css">
p
{
border:red solid thin;
outline-style:dotted;
outline-color:#00ff00;
}
</style>
</head>

<body>
<p><b>注释：</b>只有在规定了 !DOCTYPE 时，Internet Explorer 8 （以及更高版本） 才支持 outline-co
</body>
</html>
```

相关页面

CSS 教程：CSS 轮廓

CSS 参考手册：outline 属性

HTML DOM 参考手册：outlineColor 属性

CSS outline-style 属性

实例

设置轮廓的样式：

```
p
  {
  outline-style:dotted;
  }
```

浏览器支持

所有浏览器都支持 outline-style 属性。

注释：如果规定了 !DOCTYPE，则 IE8 支持 outline 属性。

定义和用法

outline-style 属性用于设置元素的整个轮廓的样式。样式不能是 none，否则轮廓不会出现。

outline （轮廓）是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。outline 属性设置元素周围的轮廓线。

注释：请始终在 outline-color 属性之前声明 outline-style 属性。元素只有获得轮廓以后才能改变其轮廓的颜色。

注释：轮廓线不会占据空间，也不一定是矩形。

默认值：	none
继承性：	no
---	---
版本：	CSS2
---	---
JavaScript 语法：	*object*.style.outlineStyle="dotted"
---	---

可能的值

值	描述
none	默认。定义无轮廓。
dotted	定义点状的轮廓。
dashed	定义虚线轮廓。
solid	定义实线轮廓。
double	定义双线轮廓。双线的宽度等同于 outline-width 的值。
groove	定义 3D 凹槽轮廓。此效果取决于 outline-color 值。
ridge	定义 3D 凸槽轮廓。此效果取决于 outline-color 值。
inset	定义 3D 凹边轮廓。此效果取决于 outline-color 值。
outset	定义 3D 凸边轮廓。此效果取决于 outline-color 值。
inherit	规定应该从父元素继承轮廓样式的设置。

TIY 实例

设置轮廓的样式

本例演示如何设置轮廓的样式。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtm
<html>
<head>
<style type="text/css">
p
{
border: red solid thin;
}
p.dotted {outline-style: dotted}
p.dashed {outline-style: dashed}
p.solid {outline-style: solid}
p.double {outline-style: double}
p.groove {outline-style: groove}
p.ridge {outline-style: ridge}
p.inset {outline-style: inset}
p.outset {outline-style: outset}
</style>
</head>
<body>

<p class="dotted">A dotted outline</p>
<p class="dashed">A dashed outline</p>
<p class="solid">A solid outline</p>
<p class="double">A double outline</p>
<p class="groove">A groove outline</p>
<p class="ridge">A ridge outline</p>
<p class="inset">An inset outline</p>
<p class="outset">An outset outline</p>

<p><b>注释：</b>只有在规定了 !DOCTYPE 时，Internet Explorer 8 （以及更高版本） 才支持 outline-st
</body>
</html>
```

相关页面

CSS 教程：CSS 轮廓

CSS 参考手册：outline 属性

HTML DOM 参考手册：outlineStyle 属性

CSS outline-width 属性

实例

设置点状轮廓的颜色：

```
p
  {
  outline-style:dotted;
  outline-width:5px;
  }
```

浏览器支持

所有浏览器都支持 outline-width 属性。

注释：如果规定了 !DOCTYPE，则 IE8 支持 outline 属性。

定义和用法

outline-width 属性设置元素整个轮廓的宽度，只有当轮廓样式不是 none 时，这个宽度才会起作用。如果样式为 none，宽度实际上会重置为 0。不允许设置负长度值。

outline（轮廓）是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。outline 属性设置元素周围的轮廓线。

注释：请始终在 outline-width 属性之前声明 outline-style 属性。元素只有获得轮廓以后才能改变其轮廓的颜色。

注释：轮廓线不会占据空间，也不一定是矩形。

默认值：	medium
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.outlineWidth="thin"

可能的值

值	描述
thin	规定细轮廓。
medium	默认。规定中等的轮廓。
thick	规定粗的轮廓。
length	允许您规定轮廓粗细的值。
inherit	规定应该从父元素继承轮廓宽度的设置。

TIY 实例

设置轮廓的宽度

本例演示如何设置轮廓的宽度。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtm
<html>
<head>
<style type="text/css">
p.one
{
border:red solid thin;
outline-style:solid;
outline-width:thin;
}
p.two
{
border:red solid thin;
outline-style:dotted;
outline-width:3px;
}
</style>
</head>
<body>

<p class="one">This is some text in a paragraph.</p>
<p class="two">This is some text in a paragraph.</p>

<p><b>注释：</b>只有在规定了 !DOCTYPE 时，Internet Explorer 8 （以及更高版本） 才支持 outline-wi
</body>
</html>
```

相关页面

CSS 教程：CSS 轮廓

CSS 参考手册：outline 属性

HTML DOM 参考手册：outlineWidth 属性

CSS3 border-bottom-left-radius 属性

实例

向 div 元素的左下角添加圆角边框：

```
div
{
border:2px solid;
border-bottom-left-radius:2em;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

IE9+、Firefox 4+、Chrome、Safari 5+ 以及 Opera 支持 border-bottom-left-radius 属性。

定义和用法

border-bottom-left-radius 属性定义左下角边框的形状。

提示：该属性允许您向元素添加圆角边框。

默认值：	0
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.borderBottomLeftRadius="5px"
---	---

语法

```
border-bottom-left-radius: _length_|_%_ [_length_|_%_];
```

注释：border-bottom-left-radius 属性的长度值和百分比值定义四分之一椭圆（定义外部边框边缘的边角形状）的半径（radii）。第一个值是水平半径，第二个值是垂直半径。如果省略第二个值，则复制第一个值。如果长度为零，则边角为方形，而不是圆形。水平半径的百分比值参考边框盒的宽度，而垂直半径的百分比值参考边框盒的高度。

值	描述	测试
length	定义左下角的形状。	测试
%	以百分比值定义左下角的形状。	测试

相关页面

CSS3 教程：CSS3 边框

CSS3 border-bottom-right-radius 属性

实例

向 div 元素的右下角添加圆角边框：

```
div
{
border:2px solid;
border-bottom-right-radius:2em;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

IE9+、Firefox 4+、Chrome、Safari 5+ 以及 Opera 支持 border-bottom-right-radius 属性。

定义和用法

border-bottom-right-radius 属性定义右下角边框的形状。

提示：该属性允许您向元素添加圆角边框。

默认值：	0
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.borderBottomRightRadius="5px"
---	---

语法

```
border-bottom-right-radius: _length_|_%_ [_length_|_%_];
```

注释：border-bottom-right-radius 属性的长度值和百分比值定义四分之一椭圆（定义外部边框边缘的边角形状）的半径（radii）。第一个值是水平半径，第二个值是垂直半径。如果省略第二个值，则复制第一个值。如果长度为零，则边角为方形，而不是圆形。水平半径的百分比值参考边框盒的宽度，而垂直半径的百分比值参考边框盒的高度。

值	描述	测试
length	定义右下角的形状。	测试
%	以百分比值定义右下角的形状。	测试

相关页面

CSS3 教程：CSS3 边框

CSS3 border-image 属性

实例

将图片规定为包围 div 元素的边框：

```
div
{
-webkit-border-image:url(border.png) 30 30 round; /* Safari 5 */
-o-border-image:url(border.png) 30 30 round; /* Opera */
border-image:url(border.png) 30 30 round;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 11, Firefox, Opera 15, Chrome 以及 Safari 6 支持 border-image 属性。

Safari 5 支持替代的 -webkit-border-image 属性。

定义和用法

border-image 属性是一个简写属性，用于设置以下属性：

- border-image-source
- border-image-slice
- border-image-width
- border-image-outset
- border-image-repeat

如果省略值，会设置其默认认值。

提示：请使用 border-image-* 属性来构造漂亮的可伸缩按钮！

默认值：	**none 100% 1 0 stretch**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.borderImage="url(border.png) 30 30 round"

可能的值

值	描述	测试
border-image-source	用在边框的图片的路径。	
border-image-slice	图片边框向内偏移。	
border-image-width	图片边框的宽度。	
border-image-outset	边框图像区域超出边框的量。	
border-image-repeat	图像边框是否应平铺(repeated)、铺满(rounded)或拉伸(stretched)。	测试

亲自试一试 - 实例

Border-image 按钮

本例演示如何通过 border-image 属性来创建按钮。

```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
border:10px solid transparent;
width:40px;
padding:5px 10px;
-moz-border-image: url(/i/border_image_button.png) 0 14 0 14 stretch; /* 老版本的 Firefox
-webkit-border-image: url(/i/border_image_button.png) 0 14 0 14 stretch; /* Safari */
-o-border-image: url(/i/border_image_button.png) 0 14 0 14 stretch; /* Opera */
border-image: url(/i/border_image_button.png) 0 14 0 14 stretch;
}
</style>
</head>
<body>

<p><b>注释：</b>Internet Explorer 不支持 border-image 属性。</p>

<div>Search</div>

<p>这是我们使用的图片：</p>
<img src="/i/border_image_button.png">

</body>
</html>
```

相关页面

CSS3 教程：CSS3 边框

CSS3 border-image-outset 属性

实例

设置 border-image-outset 属性：

```
div
{
border-image-source: url(border.png);
border-image-outset: 30 30;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Opera 不支持 border-image-outset 属性。

Internet Explorer 10 以及更早的版本不支持 border-image-outset 属性。

Safari 5 以及更早的版本不支持 border-image-outset 属性。

请参阅 border-image 属性。

定义和用法

border-image-outset 属性规定边框图像超过边框盒的量。

默认值：	0
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.borderImageOutset="30 30"
---	---

可能的值

```
border-image-outset: _length_|_number_;
```

注释：border-image-outset 属性规定边框图像超出边框盒的量。在上、右、下、左侧。如果忽略第四个值，则与第二个值相同。如果省略第三个值，则与第一个值相同。如果省略第二个值，则与第一个值相同。不允许任何负值作为 border-image-outset 值。

值	描述
length	
number	代表对应的 border-width 的倍数。

相关页面

CSS3 教程：CSS3 边框

CSS3 border-image-repeat 属性

实例

规定如何重复图像边框：

```
div
{
border-image-source: url(border.png);
border-image-repeat: round;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Opera 不支持 border-image-repeat 属性。

Internet Explorer 10 以及更早的版本不支持 border-image-repeat 属性。

Safari 5 以及更早的版本不支持 border-image-repeat 属性。

请参阅 border-image 属性。

定义和用法

border-image-repeat 属性规定图像边框是否应该被重复（repeated）、拉伸（stretched）或铺满（rounded）。

默认值：	stretch
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.borderImageRepeat="round"

可能的值

```
border-image-repeat: stretch|repeat|round;
```

注释：该属性规定如何延展和铺排边框图像的边缘和中间部分。因此，您可以规定两个值。如果省略第二个值，则采取与第一个值相同的值。

值	描述
stretch	拉伸图像来填充区域
repeat	平铺（重复）图像来填充区域。
round	类似 repeat 值。如果无法完整平铺所有图像，则对图像进行缩放以适应区域。

相关页面

CSS3 教程：CSS3 边框

CSS3 border-image-slice 属性

实例

规定图像边框的向内偏移：

```
div
{
border-image-source: url(border.png);
border-image-slice: 50% 50%;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Opera 不支持 border-image-slice 属性。

Internet Explorer 10 以及更早的版本不支持 border-image-slice 属性。

Safari 5 以及更早的版本不支持 border-image-slice 属性。

请参阅 border-image 属性。

定义和用法

border-image-slice 属性规定图像边框的向内偏移。

默认值：	100%
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.borderImageSlice="50% 50%"
---	---

语法

```
border-image-slice: _number_|_%_|fill;
```

注释：该属性规定图像的上、右、下、左侧边缘的向内偏移，图像被分割为九个区域：四个角、四条边以及一个中间区域。除非使用了关键词 fill，否则中间的图像部分会被丢弃。如果省略第四个数值/百分比，则与第二个值相同。如果省略第三个值，则与第一个值相同。如果省略第二个值，则与第一个值相同。

值	描述
number	数字值，代表图像中像素（如果是光栅图像）或矢量坐标（如果是矢量图像）。
%	相对于图像尺寸的百分比值：图像的宽度影响水平偏移，高度影响垂直偏移。
fill	保留边框图像的中间部分。

相关页面

CSS3 教程：CSS3 边框

CSS3 border-image-source 属性

实例

使用一幅图像作为围绕 div 元素的边框：

```
div
{
border-image-source: url(border.png);
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Opera 不支持 border-image-source 属性。

Internet Explorer 10 以及更早的版本不支持 border-image-source 属性。

Safari 5 以及更早的版本不支持 border-image-source 属性。

请参阅 border-image 属性。

定义和用法

border-image-source 属性规定要使用的图像，代替 border-style 属性中设置的边框样式。

提示：如果值为 "none"，或者如果图像无法显示，则使用边框样式。

默认值：	none
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.borderImageSource="url(border.png)"

语法

```
border-image-source: none|_image_;
```

值	描述
none	不使用图像。
image	用作边框的图像的路径。

相关页面

CSS3 教程：CSS3 边框

CSS3 border-image-width 属性

实例

规定图像边框的宽度：

```
div
{
border-image-source: url(border.png);
border-image-width: 30 30;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Opera 不支持 border-image-width 属性。

Internet Explorer 10 以及更早的版本不支持 border-image-width 属性。

Safari 5 以及更早的版本不支持 border-image-width 属性。

请参阅 border-image 属性。

定义和用法

border-image-width 属性规定图像边框的宽度。

默认值：	none
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.borderImageWidth="30 30"
---	---

语法

```
border-image-width: _number_|_%_|auto;
```

注释：border-image-width 属性的四个之规定将边框图像分割为九个部分的偏移。它们代表了从区域的上、右、下、左侧向内的距离。如果忽略第四个值，则与第二个值相同。如果省略第三个值，则与第一个值相同。如果省略第二个值，则与第一个值相同。不允许任何负值作为 border-image-width 值。

值	描述
number	代表对应的 border-width 的倍数。
%	参考边框图像区域的尺寸：区域的高度影响水平偏移，宽度影响垂直偏移。
auto	如果规定该属性，则宽度为对应的图像切片的固有宽度。

相关页面

CSS3 教程：CSS3 边框

CSS3 border-radius 属性

实例

向 div 元素添加圆角边框：

```
div
{
border:2px solid;
border-radius:25px;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

IE9+、Firefox 4+、Chrome、Safari 5+ 以及 Opera 支持 border-radius 属性。

定义和用法

border-radius 属性是一个简写属性，用于设置四个 border-*-radius 属性。

提示：该属性允许您为元素添加圆角边框！

默认值：	**0**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.borderRadius="5px"

语法

```
border-radius: _1-4 length_|_%_ / _1-4 length_|_%_;
```

注释：按此顺序设置每个 radii 的四个值。如果省略 bottom-left，则与 top-right 相同。如果省略 bottom-right，则与 top-left 相同。如果省略 top-right，则与 top-left 相同。

值	描述	测试
length	定义圆角的形状。	测试
%	以百分比定义圆角的形状。	测试

例子 1

```
border-radius:2em;
```

等价于：

```
border-top-left-radius:2em;
border-top-right-radius:2em;
border-bottom-right-radius:2em;
border-bottom-left-radius:2em;
```

例子 2

```
border-radius: 2em 1em 4em / 0.5em 3em;
```

等价于：

```
border-top-left-radius: 2em 0.5em;
border-top-right-radius: 1em 3em;
border-bottom-right-radius: 4em 0.5em;
border-bottom-left-radius: 1em 3em;
```

相关页面

CSS3 教程：CSS3 边框

CSS3 border-top-left-radius 属性

实例

向 div 元素的左上角添加圆角边框：

```
div
{
border:2px solid;
border-top-left-radius:2em;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

IE9+、Firefox 4+、Chrome、Safari 5+ 以及 Opera 支持 border-top-left-radius 属性。

定义和用法

border-top-left-radius 属性定义左上角边框的形状。

提示：该属性允许您向元素添加圆角边框。

默认值：	**0**
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.borderTopLeftRadius="5px"
---	---

语法

```
border-bottom-right-radius: _length_|_%_ [_length_|_%_];
```

注释：border-top-left-radius 属性的长度值和百分比值定义四分之一椭圆（定义外部边框边缘的边角形状）的半径（radii）。第一个值是水平半径，第二个值是垂直半径。如果省略第二个值，则复制第一个值。如果长度为零，则边角为方形，而不是圆形。水平半径的百分比值参考边框盒的宽度，而垂直半径的百分比值参考边框盒的高度。

值	描述	测试
length	定义左下角的形状。	测试
%	以百分比值定义左下角的形状。	测试

相关页面

CSS3 教程：CSS3 边框

CSS3 border-top-right-radius 属性

实例

向 div 元素的右上角添加圆角边框：

```
div
{
border:2px solid;
border-top-right-radius:2em;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

IE9+、Firefox 4+、Chrome、Safari 5+ 以及 Opera 支持 border-top-right-radius 属性。

定义和用法

border-top-right-radius 属性定义右下角边框的形状。

提示：该属性允许您向元素添加圆角边框。

默认值：	**0**
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.borderTopRightRadius="5px"
---	---

语法

```
border-top-right-radius: _length_|_%_ [_length_|_%_];
```

注释：border-top-right-radius 属性的长度值和百分比值定义四分之一椭圆（定义外部边框边缘的边角形状）的半径（radii）。第一个值是水平半径，第二个值是垂直半径。如果省略第二个值，则复制第一个值。如果长度为零，则边角为方形，而不是圆形。水平半径的百分比值参考边框盒的宽度，而垂直半径的百分比值参考边框盒的高度。

值	描述	测试
length	定义右上角的形状。	测试
%	以百分比值定义右上角的形状。	测试

相关页面

CSS3 教程：CSS3 边框

CSS3 box-shadow 属性

实例

向 div 元素添加 box-shadow：

```
div
{
box-shadow: 10px 10px 5px #888888;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

IE9+、Firefox 4、Chrome、Opera 以及 Safari 5.1.1 支持 box-shadow 属性。

定义和用法

box-shadow 属性向框添加一个或多个阴影。

提示：请使用 border-image-* 属性来构造漂亮的可伸缩按钮！

默认值：	none
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.boxShadow="10px 10px 5px #888888"

语法

```
box-shadow: _h-shadow_ _v-shadow_ _blur_ _spread_ _color_ inset;
```

注释：box-shadow 向框添加一个或多个阴影。该属性是由逗号分隔的阴影列表，每个阴影由 2-4 个长度值、可选的颜色值以及可选的 inset 关键词来规定。省略长度的值是 0。

值	描述	测试
h-shadow	必需。水平阴影的位置。允许负值。	测试
v-shadow	必需。垂直阴影的位置。允许负值。	测试
blur	可选。模糊距离。	测试
spread	可选。阴影的尺寸。	测试
color	可选。阴影的颜色。请参阅 CSS 颜色值。	测试
inset	可选。将外部阴影 (outset) 改为内部阴影。	测试

亲自试一试 - 实例

扔到桌子上面的图片

本例演示如何创建"宝丽来"图片，并旋转图片。

```
<!DOCTYPE html>
<html>
<head>
<style>
body
{
margin:30px;
background-color:#E9E9E9;
}

div.polaroid
{
width:294px;
padding:10px 10px 20px 10px;
border:1px solid #BFBFBF;
background-color:white;
/* Add box-shadow */
box-shadow:2px 2px 3px #aaaaaa;
}

div.rotate_left
{
float:left;
-ms-transform:rotate(7deg); /* IE 9 */
-moz-transform:rotate(7deg); /* Firefox */
-webkit-transform:rotate(7deg); /* Safari and Chrome */
-o-transform:rotate(7deg); /* Opera */
transform:rotate(7deg);
}

div.rotate_right
{
float:left;
-ms-transform:rotate(-8deg); /* IE 9 */
-moz-transform:rotate(-8deg); /* Firefox */
-webkit-transform:rotate(-8deg); /* Safari and Chrome */
-o-transform:rotate(-8deg); /* Opera */
transform:rotate(-8deg);
}
</style>
</head>
<body>

<div class="polaroid rotate_left">
<img src="/i/ballade_dream.jpg" alt="郁金香" width="284" height="213" />
<p class="caption">上海鲜花港的郁金香，花名：Ballade Dream。</p>
</div>

<div class="polaroid rotate_right">
<img src="/i/china_pavilion.jpg" alt="世博中国馆" width="284" height="213" />
<p class="caption">2010年上海世博会，中国馆。</p>
</div>

</body>
</html>
```

相关页面

CSS3 教程：CSS3 边框

Box 属性

属性	描述	CSS
overflow-x	如果内容溢出了元素内容区域，是否对内容的左/右边缘进行裁剪。	3
overflow-y	如果内容溢出了元素内容区域，是否对内容的上/下边缘进行裁剪。	3
overflow-style	规定溢出元素的首选滚动方法。	3
rotation	围绕由 rotation-point 属性定义的点对元素进行旋转。	3
rotation-point	定义距离上左边框边缘的偏移点。	3

CSS3 overflow-x 属性

实例

裁剪 div 元素中内容的左/右边缘 - 如果溢出元素的内容区域的话：

```
div
{
overflow-x:hidden;
}
```

浏览器支持

所有主流浏览器都支持 overflow-x 属性。

注释：overflow-x 属性无法在 IE8 以及更早的浏览器正确地工作。

定义和用法

overflow-x 属性规定是否对内容的左/右边缘进行裁剪 - 如果溢出元素内容区域的话。

提示：使用 overflow-y 属性来确定对上/下边缘的裁剪。

默认值：	visible
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.overflowX="scroll"

语法

```
overflow-x: visible|hidden|scroll|auto|no-display|no-content;
```

值	描述	测试
visible	不裁剪内容，可能会显示在内容框之外。	测试
hidden	裁剪内容 - 不提供滚动机制。	测试
scroll	裁剪内容 - 提供滚动机制。	测试
auto	如果溢出框，则应该提供滚动机制。	测试
no-display	如果内容不适合内容框，则删除整个框。	测试
no-content	如果内容不适合内容框，则隐藏整个内容。	测试

CSS3 overflow-y 属性

实例

裁剪 div 元素中内容的左/右边缘 - 如果溢出元素的内容区域的话：

```
div
{
overflow-y:hidden;
}
```

浏览器支持

所有主流浏览器都支持 overflow-y 属性。

注释：overflow-y 属性无法在 IE8 以及更早的浏览器正确地工作。

定义和用法

overflow-y 属性规定是否对内容的上/下边缘进行裁剪 - 如果溢出元素内容区域的话。

提示：使用 overflow-x 属性来确定对左/右边缘的裁剪。

默认值：	visible
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.overflowY="scroll"
---	---

语法

```
overflow-y: visible|hidden|scroll|auto|no-display|no-content;
```

值	描述	测试
visible	不裁剪内容，可能会显示在内容框之外。	测试
hidden	裁剪内容 - 不提供滚动机制。	测试
scroll	裁剪内容 - 提供滚动机制。	测试
auto	如果溢出框，则应该提供滚动机制。	测试
no-display	如果内容不适合内容框，则删除整个框。	测试
no-content	如果内容不适合内容框，则隐藏整个内容。	测试

CSS3 overflow-style 属性

实例

设置溢出元素的首选滚动方法：

```
div
{
overflow:auto;
overflow-style:marquee,panner;
}
```

浏览器支持

目前没有浏览器支持 overflow-style 属性。

定义和用法

overflow-style 规定溢出元素的首选滚动方法。

默认值：	**auto**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.overflowStyle="scrollbar"

语法

```
overflow-style: auto|scrollbar|panner|move|marquee;
```

注释：值既可以是 auto，或者是按照优先次序的方法列表。浏览器应该使用列表中其支持的的第一个滚动方法。

值	描述
auto	
scrollbar	为溢出元素添加滚动条。
panner	
move	用户能够直接移动元素的内容。通常，用户能够用鼠标拖动内容。
marquee	内容自主移动，不需任何用户代理对其控制。

CSS3 rotation 属性

实例

将 h1 元素旋转 180 度（从上向下）：

```
h1
{
rotation-point:50% 50%;
rotation:180deg;
}
```

浏览器支持

目前没有浏览器支持 rotation 属性。

定义和用法

rotation 属性围绕由 rotation-point 属性定义的指定点，对块级元素进行逆时针旋转。

提示：边框、内边距、内容以及背景（非固定）也会被旋转！

默认值：	**0**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.rotation="180deg"

语法

```
rotation: _angle_;
```

值	描述
angle	元素旋转角度。可能的值：0deg 到 360deg。

CSS3 rotation-point 属性

实例

将 h1 元素旋转 180 度（从上向下）：

```
h1
{
rotation-point:50% 50%;
rotation:180deg;
}
```

浏览器支持

目前没有浏览器支持 rotation-point 属性。

定义和用法

rotation-point 属性是一对值，定义从上左边框边缘进行偏移的点。

提示：rotation-point 属性需要与 rotation 属性结合使用。

默认值：	50% 50%
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.rotationPoint="25% 25%"

语法

```
rotation-point: _value_;
```

值	描述

- left top
- left center
- left bottom
- right top
- right center
- right bottom
- center top
- center center
- center bottom

| 如果只规定一个关键词，则第二个值将是 "center"。 | | *x% y%* | 百分比值，参考边框盒宽度和高度。 |

Color 属性

属性	描述	CSS
color-profile	允许使用源的颜色配置文件的默认以外的规范。	3
opacity	规定书签的级别。	3
rendering-intent	允许使用颜色配置文件渲染意图的默认以外的规范。	3

CSS3 opacity 属性

实例

设置 div 元素的不透明级别：

```
div
{
opacity:0.5;
}
```

您可以在本页底部找到更多实例。

浏览器支持

所有浏览器都支持 opacity 属性。

注释：IE8 以及更早的版本支持替代的 filter 属性。例如：filter:Alpha(opacity=50)。

定义和用法

opacity 属性设置元素的不透明级别。

默认值：	1
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.opacity=0.5

语法

```
opacity: _value_|inherit;
```

值	描述	测试
value	规定不透明度。从 0.0 （完全透明）到 1.0（完全不透明）。	测试
inherit	应该从父元素继承 opacity 属性的值。	

亲自试一试 - 实例

改变元素的不透明度

本例演示如何使用 JavaScript 来改变元素的不透明度。

```
<!DOCTYPE html>
<html>
<head>
<script>
function ChangeOpacity(x)
{
// 返回被选选项的文本
var opacity=x.options[x.selectedIndex].text;
var el=document.getElementById("p1");
if (el.style.opacity!==undefined)
  {el.style.opacity=opacity;}
else
  {alert("Your browser doesn't support this example!");}
}
</script>
</head>
<body>

<p id="p1">请从下面的例子中选择一个值，以改变此元素的不透明度。</p>
<select onchange="ChangeOpacity(this);" size="5">
  <option />0
  <option />0.2
  <option />0.5
  <option />0.8
  <option selected="selected" />1
</select>

</body>
</html>
```

相关页面

CSS 教程：CSS 图像透明度

Content for Paged Media 属性

属性	描述	CSS
bookmark-label	规定书签的标记。	3
bookmark-level	规定书签的级别。	3
bookmark-target	规定书签链接的目标。	3
float-offset	将元素放在 float 属性通常放置的位置的相反方向。	3
hyphenate-after	规定连字单词中连字符之后的最小字符数。	3
hyphenate-before	规定连字单词中连字符之前的最小字符数。	3
hyphenate-character	规定当发生断字时显示的字符串。	3
hyphenate-lines	指示元素中连续断字连线的最大数。	3
hyphenate-resource	规定帮助浏览器确定断字点的外部资源（逗号分隔的列表）。	3
hyphens	设置如何对单词进行拆分，以改善段落的布局。	3
image-resolution	规定图像的正确分辨率。	3
marks	向文档添加裁切标记或十字标记。	3
string-set	3	

CSS 尺寸属性（Dimension）

属性	描述	CSS
height	设置元素高度。	1
max-height	设置元素的最大高度。	2
max-width	设置元素的最大宽度。	2
min-height	设置元素的最小高度。	2
min-width	设置元素的最小宽度。	2
width	设置元素的宽度。	1

CSS height 属性

实例

设置段落的高度和宽度：

```
p
  {
  height:100px;
  width:100px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 height 属性。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

height 属性设置元素的高度。

说明

这个属性定义元素内容区的高度，在内容区外面可以增加内边距、边框和外边距。

行内非替换元素会忽略这个属性。

默认值：	auto
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.height="50px"

可能的值

值	描述
auto	默认。浏览器会计算出实际的高度。
length	使用 px、cm 等单位定义高度。
%	基于包含它的块级对象的百分比高度。
inherit	规定应该从父元素继承 height 属性的值。

TIY 实例

使用像素值设置图像的高度

本例演示如何使用像素值设置元素的高度。

```
<html>
<head>
<style type="text/css">
img.normal
{
height: auto
}

img.big
{
height: 160px
}

img.small
{
height: 30px
}
</style>
</head>
<body>

<img class="normal" src="/i/eg_smile.gif" />
<br />
<img class="big" src="/i/eg_smile.gif" />
<br />
<img class="small" src="/i/eg_smile.gif" />

</body>
</html>
```

使用百分比设置图像的高度

本例演示如何使用百分比值来设置元素的高度。

```
<html>
<head>
<style type="text/css">
img.normal
{
height: auto
}

img.big
{
height: 50%
}

img.small
{
height: 10%
}
</style>
</head>
<body>

<img class="normal" src="/i/eg_smile.gif" />
<br />
<img class="big" src="/i/eg_smile.gif" />
<br />
<img class="small" src="/i/eg_smile.gif" />

</body>
</html>
```

相关页面

CSS 教程：CSS 尺寸

CSS 教程：CSS 框模型概述

CSS 参考手册：width 属性

HTML DOM 参考手册：height 属性

CSS max-height 属性

实例

设置段落的最大高度：

```
p
  {
  max-height:100px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 max-height 属性。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

max-height 属性设置元素的最大高度。

说明

该属性值会对元素的高度设置一个最高限制。因此，元素可以比指定值矮，但不能比其高。不允许指定负值。

注释：max-height 属性不包括外边距、边框和内边距。

默认值：	none
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.maxHeight="50px"

可能的值

值	描述
none	默认。定义对元素被允许的最大高度没有限制。
length	定义元素的最大高度值。
%	定义基于包含它的块级对象的百分比最大高度。
inherit	规定应该从父元素继承 max-height 属性的值。

TIY 实例

设置元素的最大高度

本例演示如何设置一个元素的最大高度。

```
<html>
<head>
<style type="text/css">
p
{
max-height: 10px
}
</style>
</head>
<body>

<p>这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。</p>

<img src="/i/eg_smile.gif" />

</body>
</html>
```

相关页面

CSS 教程：CSS 尺寸

CSS 教程：min-height 属性

HTML DOM 参考手册：maxHeight 属性

CSS max-width 属性

实例

设置段落的最大宽度：

```
p
  {
  max-width:100px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 max-width 属性。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

max-width 定义元素的最大宽度。

说明

该属性值会对元素的宽度设置一个最高限制。因此，元素可以比指定值窄，但不能比其宽。不允许指定负值。

默认值：	none
继承性：	no
---	---
版本：	CSS2
---	---
JavaScript 语法：	*object*.style.maxWidth="50px"
---	---

可能的值

值	描述
none	默认。定义对元素的最大宽度没有限制。
length	定义元素的最大宽度值。
%	定义基于包含它的块级对象的百分比最大宽度。
inherit	规定应该从父元素继承 max-width 属性的值。

TIY 实例

使用像素值来设置元素的最大宽度

本例演示如何使用像素值来设置元素的最大高度。

```
<html>
<head>
<style type="text/css">
p
{
max-width: 300px
}
</style>
</head>
<body>

<p>这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。</p>

</body>
</html>
```

使用百分比来设置元素的最大宽度

本例演示如何使用百分比值来设置元素的最大高度。

```
<html>
<head>
<style type="text/css">
p
{
max-width: 50%
}
</style>
</head>
<body>

<p>这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 尺寸

CSS 参考手册：CSS min-width 属性

HTML DOM 参考手册：maxWidth 属性

CSS min-height 属性

实例

设置段落的最小高度：

```
p
  {
  min-height:100px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 min-height 属性。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

min-height 属性设置元素的最小高度。

说明

该属性值会对元素的高度设置一个最低限制。因此，元素可以比指定值高，但不能比其矮。不允许指定负值。

默认值：	0
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.minHeight="50px"

可能的值

值	描述
length	定义元素的最小高度。默认值是 0。
%	定义基于包含它的块级对象的百分比最小高度。
inherit	规定应该从父元素继承 min-height 属性的值。

TIY 实例

使用像素值来设置元素的最小高度

本例演示如何使用像素值来设置元素的最小高度。

```
<html>
<head>
<style type="text/css">
p
{
min-height: 100px
}
</style>
</head>
<body>

<p>这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。</p>

<img src="/i/eg_smile.gif" />

</body>
</html>
```

相关页面

CSS 教程：CSS 尺寸

CSS 教程：max-height 属性

HTML DOM 参考手册：minHeight 属性

CSS min-width 属性

实例

设置段落的最小宽度：

```
p
  {
  min-width:100px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 min-width 属性。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

min-width 属性设置元素的最小宽度。

说明

该属性值会对元素的宽度设置一个最小限制。因此，元素可以比指定值宽，但不能比其窄。不允许指定负值。

默认值：	none
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.minWidth="50px"

可能的值

值	描述
length	定义元素的最小宽度值。默认值：取决于浏览器。
%	定义基于包含它的块级对象的百分比最小宽度。
inherit	规定应该从父元素继承 min-width 属性的值。

TIY 实例

使用像素值来设置元素的最小宽度

本例演示如何使用像素值来设置元素的最小宽度。

```
<html>
<head>
<style type="text/css">
p
{
min-width: 1000px
}
</style>
</head>
<body>

<p>这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。</p>

<img src="/i/eg_smile.gif" />

</body>
</html>
```

使用百分比来设置元素的最小宽度

本例演示如何使用百分比值来设置元素的最小宽度。

```
<html>
<head>
<style type="text/css">
p
{
min-width: 200%
}
</style>
</head>
<body>

<p>这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。
这是一些文本。这是一些文本。这是一些文本。</p>

<img src="/i/eg_smile.gif" />

</body>
</html>
```

相关页面

CSS 教程：CSS 尺寸

CSS 参考手册：CSS max-width 属性

HTML DOM 参考手册：minWidth 属性

CSS width 属性

实例

设置段落的高度和宽度：

```
p
  {
  height:100px;
  width:100px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 width 属性。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

width 属性设置元素的宽度。

说明

这个属性定义元素内容区的宽度，在内容区外面可以增加内边距、边框和外边距。

行内非替换元素会忽略这个属性。

默认值：	auto
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.width="50px"

可能的值

值	描述
auto	默认值。浏览器可计算出实际的宽度。
length	使用 px、cm 等单位定义宽度。
%	定义基于包含块（父元素）宽度的百分比宽度。
inherit	规定应该从父元素继承 width 属性的值。

TIY 实例

使用像素值来设置元素的宽度

本例演示如何使用像素值来设置元素的宽度。

```
<html>
<head>
<style type="text/css">
img
{
width: 300px
}
</style>
</head>
<body>

<img src="/i/eg_smile.gif" />

</body>
</html>
```

使用百分比来设置元素的宽度

本例演示如何使用百分比值来设置元素的宽度。

```
<html>
<head>
<style type="text/css">
img
{
width: 50%
}
</style>
</head>
<body>

<img src="/i/eg_smile.gif" />

</body>
</html>
```

相关页面

CSS 教程：CSS 尺寸

CSS 教程：CSS 框模型概述

CSS 参考手册：height 属性

HTML DOM 参考手册：width 属性

可伸缩框属性（Flexible Box）

属性	描述	CSS
box-align	规定如何对齐框的子元素。	3
box-direction	规定框的子元素的显示方向。	3
box-flex	规定框的子元素是否可伸缩。	3
box-flex-group	将可伸缩元素分配到柔性分组。	3
box-lines	规定当超出父元素框的空间时，是否换行显示。	3
box-ordinal-group	规定框的子元素的显示次序。	3
box-orient	规定框的子元素是否应水平或垂直排列。	3
box-pack	规定水平框中的水平位置或者垂直框中的垂直位置。	3

CSS3 box-align 属性

实例

通过使用 box-align and box-pack 属性，居中 div 框的子元素：

```
div
{
width:350px;
height:100px;
border:1px solid black;

/* Firefox */
display:-moz-box;
-moz-box-pack:center;
-moz-box-align:center;

/* Safari、Opera 以及 Chrome */
display:-webkit-box;
-webkit-box-pack:center;
-webkit-box-align:center;

/* W3C */
display:box;
box-pack:center;
box-align:center;
}
```

页面底部有更多实例。

浏览器支持

目前没有浏览器支持 box-align 属性。

Firefox 支持替代的 -moz-box-align 属性。

Safari、Opera 以及 Chrome 支持替代的 -webkit-box-align 属性。

定义和用法

box-align 属性规定如何对齐框的子元素。

默认值：	**stretch**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.boxAlign="center"

语法

```
box-align: start|end|center|baseline|stretch;
```

值	描述	测试
start		

对于正常方向的框，每个子元素的上边缘沿着框的顶边放置。

对于反方向的框，每个子元素的下边缘沿着框的底边放置。

| 测试 | | end |

对于正常方向的框，每个子元素的下边缘沿着框的底边放置。

对于反方向的框，每个子元素的上边缘沿着框的顶边放置。

| 测试 | | center | 均等地分割多余的空间，一半位于子元素之上，另一半位于子元素之下。 | 测试 | | baseline | 如果 box-orient 是inline-axis或horizontal，所有子元素均与其基线对齐。 | 测试 | | stretch | 拉伸子元素以填充包含块 |

亲自试一试 - 实例

改变元素的 box-align 值

该例演示如何使用 JavaScript 来改变元素的 box-align 值。

```
<!DOCTYPE html>
<html>
<head>
<style>
.box
{
display:-moz-box;
display:-webkit-box;
display:box;
width:200px;
height:100px;
border:2px solid red;
}
</style>
<script>
function ChangeBoxAlign(x)
{
// Returns the selected option's text
var boxAlign=x.options[x.selectedIndex].text;
var div=document.getElementById("myDiv");
if (div.style.MozBoxAlign!==undefined)
  {
  div.style.MozBoxAlign=boxAlign;
  }
else if (div.style.webkitBoxAlign!==undefined)
  {
  div.style.webkitBoxAlign=boxAlign;
  }
else
  {
  alert("Your browser doesn't support this example!");
  }
}
</script>
</head>
<body>
<div class="box" id="myDiv">
  <b>第一个子元素</b>
  <i>第二个子元素</i>
</div>

<select onchange="ChangeBoxAlign (this);" size="6">
  <option selected="selected" />baseline
  <option />center
  <option />end
  <option />inherit
  <option />start
  <option />stretch
</select>

<p><b>注释：</b>本例在 Internet Explorer 和 Opera 中无效。</p>

</body>
</html>
```

CSS3 box-direction 属性

实例

以反方向显示 div 框的子元素：

```
div
{
width:350px;
height:100px;
border:1px solid black;

/* Firefox */
display:-moz-box;
-moz-box-direction:reverse;

/* Safari、Opera 以及 Chrome */
display:-webkit-box;
-webkit-box-direction:reverse;

/* W3C */
display:box;
box-direction:reverse;
}
```

浏览器支持

目前没有浏览器支持 box-direction 属性。

Firefox 支持替代的 -moz-box-direction 属性。

Safari、Opera 以及 Chrome 支持替代的 -webkit-box-direction 属性。

定义和用法

box-direction 属性规定框元素的子元素以什么方向来排列。

默认值：	normal
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.boxDirection="reverse"

语法

```
box-direction: normal|reverse|inherit;
```

值	描述	测试
normal	以默认方向显示子元素。	测试
reverse	以反方向显示子元素。	测试
inherit	应该从子元素继承 box-direction 属性的值	

CSS3 box-flex 属性

实例

定义两个可伸缩的 p 元素。如果父元素的总宽度是 300 像素，则 #p1 的宽度是 100 像素，而 #p2 的宽度是 200 像素：

```
#p1
{
-moz-box-flex:1.0; /* Firefox */
-webkit-box-flex:1.0; /* Safari 和 Chrome */
box-flex:1.0;
border:1px solid red;
}

#p2
{
-moz-box-flex:2.0; /* Firefox */
-webkit-box-flex:2.0; /* Safari 和 Chrome */
box-flex:2.0;
border:1px solid blue;
}
```

浏览器支持

目前没有浏览器支持 box-flex 属性。

Firefox 支持替代的 -moz-box-flex 属性。

Safari、Opera 以及 Chrome 支持替代的 -webkit-box-flex 属性。

定义和用法

box-flex 属性规定框的子元素是否可伸缩其尺寸。

提示：可伸缩元素能够随着框的缩小或扩大而缩写或放大。只要框中有多余的空间，可伸缩元素就会扩展来填充这些空间。

默认值：	**0.0**（指示该元素不可伸缩）
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.boxFlex=2.0

语法

```
box-flex: _value_;
```

值	描述
value	元素的可伸缩行。柔性是相对的，例如 box-flex 为 2 的子元素两倍于 box-flex 为 1 的子元素。

CSS3 box-flex-group 属性

浏览器支持

目前没有浏览器支持 box-flex-group 属性。

定义和用法

box-flex-group 属性用于向柔性分组分配可伸缩元素。

提示：可伸缩元素能够随着框的缩小和扩大而伸缩。

默认值：	**1**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.boxFlexGroup=2

语法

```
box-flex-group: _integer_;
```

值	描述
integer	一个整数。（第一个柔性分组是 1，后面的柔性分组是更大的值）。

CSS3 box-lines 属性

实例

规定允许 div 扩展为多行：

```
div
{
display:box;
box-orient:horizontal;
box-lines:multiple;
width:200px;
}
```

浏览器支持

目前没有浏览器支持 box-lines 属性。

定义和用法

box-lines 属性规定如果列超出了父框中的空间，是否要换行显示。

提示：默认地，水平框会在单独的行中排列其子元素，而垂直框会在单独的列中排列其子元素。

默认值：	single
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.boxLines="multiple"

语法

```
box-lines: single|multiple;
```

值	描述
single	所有子元素会被放置在单独的行或列中。（无法匹配的元素会被视为溢出）。
multiple	允许框扩展为多行，以容纳其所有子元素。

CSS3 box-ordinal-group 属性

实例

规定框中子元素的显示次序：

```
.box
{
display:-moz-box; /* Firefox */
display:-webkit-box; /* Safari 和 Chrome */
display:box;
border:1px solid black;
}
.ord1
{
margin:5px;
-moz-box-ordinal-group:1; /* Firefox */
-webkit-box-ordinal-group:1; /* Safari 和 Chrome */
box-ordinal-group:1;
}
.ord2
{
margin:5px;
-moz-box-ordinal-group:2; /* Firefox */
-webkit-box-ordinal-group:2; /* Safari 和 Chrome */
box-ordinal-group:2;
}
```

浏览器支持

目前没有浏览器支持 box-ordinal-group 属性。

Firefox 支持替代的 -moz-box-ordinal-group 属性。

Safari 和 Chrome 支持替代的 -webkit-box-ordinal-group 属性。

定义和用法

box-ordinal-group 属性规定框中子元素的显示次序。

值更低的元素会显示在值更高的元素前面显示。

注释：分组值相同的元素，它们的显示次序取决于其源次序。

默认值：	1
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.boxOrdinalGroup=2

语法

```
box-ordinal-group: _integer_;
```

值	描述	测试
integer	一个整数，指示子元素的显示次序。	测试

CSS3 box-orient 属性

实例

水平排列 div 元素的子元素：

```
div
{
width:350px;
height:100px;
border:1px solid black;

/* Firefox */
display:-moz-box;
-moz-box-orient:horizontal;

/* Safari、Opera 以及 Chrome */
display:-webkit-box;
-webkit-box-orient:horizontal;

/* W3C */
display:box;
box-orient:horizontal;
}
```

浏览器支持

目前没有浏览器支持 box-orient 属性。

Firefox 支持替代的 -moz-box-orient 属性。

Safari、Opera 以及 Chrome 支持替代的 -webkit-box-orient 属性。

定义和用法

box-orient 属性规定框的子元素应该被水平或垂直排列。

提示：水平框中的子元素从左向右进行显示，而垂直框的子元素从上向下进行显示。不过，box-direction 和 box-ordinal-group 能够改变这种顺序。

默认值：	**inline-axis**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.boxOrient="vertical"

语法

```
box-orient: horizontal|vertical|inline-axis|block-axis|inherit;
```

值	描述	测试
horizontal	在水平行中从左向右排列子元素。	测试
vertical	从上向下垂直排列子元素。	测试
inline-axis	沿着行内轴来排列子元素（映射为 horizontal）。	测试
block-axis	沿着块轴来排列子元素（映射为 vertical）。	测试
inherit	应该从父元素继承 box-orient 属性的值。	

CSS3 box-pack 属性

实例

通过一起使用 box-align 和 box-pack 属性，对 div 框的子元素进行居中：

```
div
{
width:350px;
height:100px;
border:1px solid black;

/* Firefox */
display:-moz-box;
-moz-box-pack:center;
-moz-box-align:center;

/* Safari、Opera 以及 Chrome */
display:-webkit-box;
-webkit-box-pack:center;
-webkit-box-align:center;

/* W3C */
display:box;
box-pack:center;
box-align:center;
}
```

浏览器支持

目前没有浏览器支持 box-pack 属性。

Firefox 支持替代的 -moz-box-pack 属性。

Safari、Opera 以及 Chrome 支持替代的 -webkit-box-pack 属性。

定义和用法

box-pack 属性规定当框大于子元素的尺寸，在何处放置子元素。

该属性规定水平框中的水平位置，以及垂直框中的垂直位置。

默认值：	start
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.boxPack="center"

语法

```
box-pack: start|end|center|justify;
```

值	描述	测试
start		

对于正常方向的框，首个子元素的左边缘被放在左侧（最后的子元素后是所有剩余的空间）

对于相反方向的框，最后子元素的右边缘被放在右侧（首个子元素前是所有剩余的空间）

| 测试 | | end |

对于正常方向的框，最后子元素的右边缘被放在右侧（首个子元素前是所有剩余的空间）。

对于相反方向的框，首个子元素的左边缘被放在左侧（最后子元素后是所有剩余的空间）。 | 测试 | | center | 均等地分割多余空间，其中一半空间被置于首个子元素前，另一半被置于最后一个子元素后 | 测试 | | justify | 在每个子元素之间分割多余的空间（首个子元素前和最后一个子元素后没有多余的空间）。 | 测试 |

CSS 字体属性（Font）

属性	描述	CSS
font	在一个声明中设置所有字体属性。	1
font-family	规定文本的字体系列。	1
font-size	规定文本的字体尺寸。	1
font-size-adjust	为元素规定 aspect 值。	2
font-stretch	收缩或拉伸当前的字体系列。	2
font-style	规定文本的字体样式。	1
font-variant	规定是否以小型大写字母的字体显示文本。	1
font-weight	规定字体的粗细。	1

CSS font 属性

实例

在一个声明中设置所有字体属性：

```
p.ex1
  {
  font:italic arial,sans-serif;
  }

p.ex2
  {
  font:italic bold 12px/20px arial,sans-serif;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 font 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

font 简写属性在一个声明中设置所有字体属性。

注释：此属性也有第六个值："line-height"，可设置行间距。

说明

这个简写属性用于一次设置元素字体的两个或更多方面。使用 icon 等关键字可以适当地设置元素的字体，使之与用户计算机环境中的某个方面一致。注意，如果没有使用这些关键词，至少要指定字体大小和字体系列。

可以按顺序设置如下属性：

- font-style
- font-variant
- font-weight
- font-size/line-height

- font-family

可以不设置其中的某个值，比如 font:100% verdana; 也是允许的。未设置的属性会使用其默认值。

默认值：	***not specified***
继承性：	yes
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.font="italic small-caps bold 12px arial,sans-serif"
---	---

可能的值

值	描述
font-style	规定字体样式。参阅：font-style 中可能的值。
font-variant	规定字体异体。参阅：font-variant 中可能的值。
font-weight	规定字体粗细。参阅：font-weight 中可能的值。
font-size/line-height	规定字体尺寸和行高。参阅：font-size 和 line-height 中可能的值。
font-family	规定字体系列。参阅：font-family 中可能的值。
caption	定义被标题控件（比如按钮、下拉列表等）使用的字体。
icon	定义被图标标记使用的字体。
menu	定义被下拉列表使用的字体。
message-box	定义被对话框使用的字体。
small-caption	caption 字体的小型版本。
status-bar	定义被窗口状态栏使用的字体。

TIY 实例

所有字体属性在一个声明之内

本例演示如何使用简写属性将字体属性设置在一个声明之内。

```
<html>
<head>
<style type="text/css">
p.ex1
{
font:italic arial,sans-serif;
}

p.ex2
{
font:italic bold 12px/30px arial,sans-serif;
}
</style>
</head>

<body>
<p class="ex1">This is a paragraph. This is a paragraph. This is a paragraph. This is a p
<p class="ex2">This is a paragraph. This is a paragraph. This is a paragraph. This is a p
</body>
</html>
```

设置使用 "caption" 值的段落字体

本例演示如何设置使用 "caption" 值的段落字体。

```
<html>
<body>

<p>This is a normal paragraph</p>
<p style="font: caption">This is a paragraph with a "caption" font</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 字体

HTML DOM 参考手册：font 属性

CSS font-family 属性

实例

为段落设置字体：

```
p
  {
  font-family:"Times New Roman",Georgia,Serif;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 font-family 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

font-family 规定元素的字体系列。

font-family 可以把多个字体名称作为一个“回退”系统来保存。如果浏览器不支持第一个字体，则会尝试下一个。也就是说，font-family 属性的值是用于某个元素的字体族名称或/及类族名称的一个优先表。浏览器会使用它可识别的第一个值。

有两种类型的字体系列名称：

- 指定的系列名称：具体字体的名称，比如："times"、"courier"、"arial"。
- 通常字体系列名称：比如："serif"、"sans-serif"、"cursive"、"fantasy"、"monospace"

提示：使用逗号分割每个值，并始终提供一个类族名称作为最后的选择。

注意：使用某种特定的字体系列（Geneva）完全取决于用户机器上该字体系列是否可用；这个属性没有指示任何字体下载。因此，强烈推荐使用一个通用字体系列名作为后路。

默认值：	***not specified***
继承性：	yes
版本：	CSS1
JavaScript 语法：	*object*.style.fontFamily="arial,sans-serif"

可能的值

值	描述

- *family-name*
- *generic-family*

  |

用于某个元素的字体族名称或/及类族名称的一个优先表。

默认值：取决于浏览器。

| | inherit | 规定应该从父元素继承字体系列。 |

TIY 实例

设置文本的字体

本例演示如何设置文本字体。

```
<html>
<head>
<style type="text/css">
p.serif{font-family:"Times New Roman",Georgia,Serif}
p.sansserif{font-family:Arial,Verdana,Sans-serif}
</style>
</head>

<body>
<h1>CSS font-family</h1>
<p class="serif">This is a paragraph, shown in the Times New Roman font.</p>
<p class="sansserif">This is a paragraph, shown in the Arial font.</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 字体

CSS 参考手册：CSS font 属性

HTML DOM 参考手册：font 属性

CSS font-size 属性

实例

设置不同的 HTML 元素的尺寸：

```
h1 {font-size:250%;}
h2 {font-size:200%;}
p {font-size:100%}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 font-size 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

font-size 属性可设置字体的尺寸。

说明

该属性设置元素的字体大小。注意，实际上它设置的是字体中字符框的高度；实际的字符字形可能比这些框高或矮（通常会矮）。

各关键字对应的字体必须比一个最小关键字相应字体要高，并且要小于下一个最大关键字对应的字体。

默认值：	medium
继承性：	yes
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.fontSize="larger"
---	---

可能的值

值	描述

- xx-small
- x-small
- small
- medium
- large
- x-large
- xx-large

|

把字体的尺寸设置为不同的尺寸，从 xx-small 到 xx-large。

默认值：medium。

| | smaller | 把 font-size 设置为比父元素更小的尺寸。 | | larger | 把 font-size 设置为比父元素更大的尺寸。 | | *length* | 把 font-size 设置为一个固定的值。 | | *%* | 把 font-size 设置为基于父元素的一个百分比值。 | | inherit | 规定应该从父元素继承字体尺寸。 |

TIY 实例

设置字体尺寸

本例演示如何设置字体尺寸。

```
<html>
<head>
<style type="text/css">
h1 {font-size: 300%}
h2 {font-size: 200%}
p {font-size: 100%}
</style>
</head>

<body>
<h1>This is header 1</h1>
<h2>This is header 2</h2>
<p>This is a paragraph</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 字体

CSS 参考手册：CSS font 属性

HTML DOM 参考手册：fontSize 属性

CSS font-size-adjust 属性

实例

设置不同的 HTML 元素的 font-size-adjust 属性：

```
h1
  {
  font-size-adjust:0.58;
  }

p
  {
  font-size-adjust:0.60;
  }
```

浏览器支持

Internet Explorer 不支持 font-size-adjust 属性。

定义和用法

font-size-adjust 属性为某个元素规定一个 aspect 值，这样就可以保持首选字体的 x-height。

说明

字体的小写字母 "x" 的高度与 "font-size" 高度之间的比率被称为一个字体的 aspect 值。当字体拥有高的 aspect 值时，那么当此字体被设置为很小的尺寸时会更易阅读。举例：Verdana 的 aspect 值是 0.58（意味着当字体尺寸为 100px 时，它的 x-height 是 58px ）。Times New Roman 的 aspect 值是 0.46。这就意味着 Verdana 在小尺寸时比 Times New Roman 更易阅读。

默认值：	**none**
继承性：	yes
版本：	CSS2
JavaScript 语法：	*object*.style.fontSizeAdjust="0.70"

可能的值

值	描述
none	默认。如果此字体不可用，则不保持此字体的 x-height。
number	

定义字体的 aspect 值比率。

可使用的公式：

首选字体的字体尺寸 * （font-size-adjust 值 / 可用字体的 aspect 值）=可应用到可用字体的字体尺寸

举例：

如果 14px 的 Verdana（aspect 值是 0.58）不可用，但是某个可用的字体的 aspect 值是 0.46，那么替代字体的尺寸将是 14 * (0.58/0.46) = 17.65px。

|

TIY 实例

使用 font-size-adjust 设置字体尺寸

本例演示如何使用 font-size-adjust 设置字体尺寸。

```
<html>
<head>
<style type="text/css">
h1 {font-size-adjust: 0.50}
h2 {font-size-adjust: 0.40}
p {font-size-adjust: 0.60}
</style>
</head>
<body>

<h1>This is header 1</h1>
<h2>This is header 2</h2>
<p>This is a paragraph</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 字体

CSS 参考手册：CSS font 属性

HTML DOM 参考手册：fontSizeAdjust 属性

CSS font-stretch 属性

实例

设置 HTML 元素的 font-stretch 属性：

```
h1
  {
  font-stretch:ultra-condensed;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都不支持 font-stretch 属性。

定义和用法

font-stretch 属性可对当前的 font-family 进行伸缩变形。

默认值：	normal
继承性：	yes
版本：	CSS2
JavaScript 语法：	*object*.style.fontStretch="ultra-expanded"

可能的值

值	描述
normal	默认值。把缩放比例设置为标准。
wider	把伸展比例设置为更进一步的伸展值
narrower	把收缩比例设置为更进一步的收缩值

- ultra-condensed
- extra-condensed
- condensed
- semi-condensed
- semi-expanded
- expanded
- extra-expanded
- ultra-expanded

|

设置 font-family 的缩放比例。

"ultra-condensed" 是最宽的值，而 "ultra-expanded" 是最窄的值。

|

相关页面

CSS 教程：CSS 字体

CSS 参考手册：CSS font 属性

HTML DOM 参考手册：fontStretch 属性

CSS font-style 属性

实例

为三个段落设置不同的字体样式：

```
p.normal {font-style:normal;}
p.italic {font-style:italic;}
p.oblique {font-style:oblique;}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 font-style 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

font-style 属性定义字体的风格。

说明

该属性设置使用斜体、倾斜或正常字体。斜体字体通常定义为字体系列中的一个单独的字体。理论上讲，用户代理可以根据正常字体计算一个斜体字体。

默认值：	normal
继承性：	yes
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.fontStyle="italic"
---	---

可能的值

值	描述
normal	默认值。浏览器显示一个标准的字体样式。
italic	浏览器会显示一个斜体的字体样式。
oblique	浏览器会显示一个倾斜的字体样式。
inherit	规定应该从父元素继承字体样式。

TIY 实例

设置字体风格

本例演示如何设置字体风格。

```
<html>
<head>
<style type="text/css">
p.normal {font-style:normal}
p.italic {font-style:italic}
p.oblique {font-style:oblique}
</style>
</head>

<body>
<p class="normal">This is a paragraph, normal.</p>
<p class="italic">This is a paragraph, italic.</p>
<p class="oblique">This is a paragraph, oblique.</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 字体

CSS 参考手册：CSS font 属性

HTML DOM 参考手册：fontStyle 属性

CSS font-variant 属性

实例

把段落设置为小型大写字母字体：

```
p.small
  {
  font-variant:small-caps;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 font-variant 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

font-variant 属性设置小型大写字母的字体显示文本，这意味着所有的小写字母均会被转换为大写，但是所有使用小型大写字体的字母与其余文本相比，其字体尺寸更小。

说明

该属性主要用于定义小型大写字母文本。理论上，用户代理可以根据正常字体计算出小型大写字母字体。

默认值：	**normal**
继承性：	yes
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.fontVariant="small-caps"
---	---

可能的值

值	描述
normal	默认值。浏览器会显示一个标准的字体。
small-caps	浏览器会显示小型大写字母的字体。
inherit	规定应该从父元素继承 font-variant 属性的值。

TIY 实例

设置字体的异体

本例演示如何设置字体的异体。

```
<html>
<head>
<style type="text/css">
p.normal {font-variant: normal}
p.small {font-variant: small-caps}
</style>
</head>

<body>
<p class="normal">This is a paragraph</p>
<p class="small">This is a paragraph</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 字体

CSS 参考手册：CSS font 属性

HTML DOM 参考手册：fontVariant 属性

CSS font-weight 属性

实例

设置三个段落的字体的粗细：

```
p.normal {font-weight:normal;}
p.thick {font-weight:bold;}
p.thicker {font-weight:900;}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 font-weight 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

font-weight 属性设置文本的粗细。

说明

该属性用于设置显示元素的文本中所用的字体加粗。数字值 400 相当于 关键字 normal，700 等价于 bold。每个数字值对应的字体加粗必须至少与下一个最小数字一样细，而且至少与下一个最大数字一样粗。

默认值：	normal
继承性：	yes
版本：	CSS1
JavaScript 语法：	*object*.style.fontWeight="900"

可能的值

值	描述
normal	默认值。定义标准的字符。
bold	定义粗体字符。
bolder	定义更粗的字符。
lighter	定义更细的字符。

- 100
- 200
- 300
- 400
- 500
- 600
- 700
- 800
- 900

| 定义由粗到细的字符。400 等同于 normal，而 700 等同于 bold。 | | inherit | 规定应该从父元素继承字体的粗细。 |

TIY 实例

设置字体的粗细

本例演示如何设置字体的粗细。

```
<html>
<head>
<style type="text/css">
p.normal {font-weight: normal}
p.thick {font-weight: bold}
p.thicker {font-weight: 900}
</style>
</head>

<body>
<p class="normal">This is a paragraph</p>

<p class="thick">This is a paragraph</p>

<p class="thicker">This is a paragraph</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 字体

CSS 参考手册：CSS font 属性

HTML DOM 参考手册：fontWeight 属性

内容生成（Generated Content）

属性	描述	CSS
content	与 :before 以及 :after 伪元素配合使用，来插入生成内容。	2
counter-increment	递增或递减一个或多个计数器。	2
counter-reset	创建或重置一个或多个计数器。	2
quotes	设置嵌套引用的引号类型。	2
crop	允许被替换元素仅仅是对象的矩形区域，而不是整个对象。	3
move-to	从流中删除元素，然后在文档中后面的点上重新插入。	3
page-policy	确定元素基于页面的 occurrence 应用于计数器还是字符串值。	3

CSS content 属性

实例

下面的例子在每个链接后插入括号中的 URL：

```
a:after
  {
  content: " (" attr(href) ")";
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 content 属性。

注释：如果已规定 !DOCTYPE，那么 Internet Explorer 8 （以及更高版本）支持 content 属性。

定义和用法

content 属性与 :before 及 :after 伪元素配合使用，来插入生成内容。

说明

该属性用于定义元素之前或之后放置的生成内容。默认地，这往往是行内内容，不过该内容创建的框类型可以用属性 display 控制。

默认值：	**normal**
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.content="url(beep.wav)"

可能的值

值	描述
none	
normal	
content specifications	
inherit	规定应该从父元素继承 content 属性的值。

相关页面

CSS 参考手册：CSS :before 伪元素

CSS 参考手册：CSS :after 伪元素

HTML DOM 参考手册：content 属性

CSS counter-increment 属性

实例

对部分和子部分进行编号（比如 "Section 1"、"1.1"、"1.2"）的方法：

```
body
  {
  counter-reset:section;
  }

h1
  {
  counter-reset:subsection;
  }

h1:before
  {
  content:"Section " counter(section) ". ";
  counter-increment:section;
  }

h2:before
  {
  counter-increment:subsection;
  content:counter(section) "." counter(subsection) " ";
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 counter-increment 属性。

注释：如果已规定 !DOCTYPE，那么 Internet Explorer 8 （以及更高版本）支持 counter-increment 属性。

定义和用法

counter-increment 属性设置某个选取器每次出现的计数器增量。默认增量是 1。

说明

利用这个属性，计数器可以递增（或递减）某个值，这可以是正值或负值。如果没有提供 number 值，则默认为 1。

注释：如果使用了 "display: none"，则无法增加计数。如使用 "visibility: hidden"，则可增加计数。

默认值：	**none**
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.counterIncrement="subsection"

可能的值

值	描述
none	默认。选择器无计数器增量。
id number	

id 定义将增加计数的选择器、id 或 class。

number 定义增量。number 可以是正数、零或者负数。

| | inherit | 规定应该从父元素继承 counter-increment 属性的值。 |

相关页面

CSS 参考手册：CSS :before 伪元素

CSS 参考手册：CSS :after 伪元素

CSS 参考手册：content 属性

CSS 参考手册：counter-reset 属性

CSS counter-reset 属性

实例

对部分和子部分进行编号（比如 "Section 1"、"1.1"、"1.2"）的方法：

```
body
  {
  counter-reset:section;
  }

h1
  {
  counter-reset:subsection;
  }

h1:before
  {
  content:"Section " counter(section) ". ";
  counter-increment:section;
  }

h2:before
  {
  counter-increment:subsection;
  content:counter(section) "." counter(subsection) " ";
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 counter-reset 属性。

注释：如果已规定 !DOCTYPE，那么 Internet Explorer 8 （以及更高版本）支持 counter-reset 属性。

定义和用法

counter-reset 属性设置某个选择器出现次数的计数器的值。默认为 0。

利用这个属性，计数器可以设置或重置为任何值，可以是正值或负值。如果没有提供 number，则默认为 0。

注释：如果使用 "display: none"，则无法重置计数器。如果使用 "visibility: hidden"，则可以重置计数器。

默认值：	**none**
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.counterReset="subsection"

可能的值

值	描述
none	默认。不能对选择器的计数器进行重置。
id number	

id 定义重置计数器的选择器、id 或 class。

number 可设置此选择器出现次数的计数器的值。可以是正数、零或负数。

| | inherit | 规定应该从父元素继承 counter-reset 属性的值。 |

相关页面

CSS 参考手册：CSS :before 伪元素

CSS 参考手册：CSS :after 伪元素

CSS 参考手册：content 属性

CSS 参考手册：counter-increment 属性

CSS quotes 属性

实例

```
q:lang(en)
  {
  quotes: '"' '"' "'" "'";
  }
```

HTML 代码：

```
<html lang="en">
<head>
</head>
<body>
<p><q>This is a <q>big</q> quote</q></p>
</body>
</html>
```

输出：

```
"This is a 'big' quote"
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 quotes 属性。

注释：如果已规定 !DOCTYPE，那么 Internet Explorer 8 （以及更高版本）支持 quotes 属性。

定义和用法

quotes 属性设置嵌套引用（embedded quotation）的引号类型。

默认值：	**not specified**
继承性：	yes
版本：	CSS2
JavaScript 语法：	*object*.style.quotes="none"

可能的值

值	描述
none	规定 "content" 属性的 "open-quote" 和 "close-quote" 的值不会产生任何引号。
string string string string	

定义要使用的引号。

前两个值规定第一级引用嵌套，后两个值规定下一级引号嵌套。

| | inherit | 规定应该从父元素继承 quotes 属性的值。 |

引号字符

Result	Description	Entity Number
"	double quote	"
'	single quote	'
‹	single, left angle quote	‹
›	single, right angle quote	›
«	double, left angle quote	«
»	double, right angle quote	»
‘	left quote (single high-6)	‘
’	right quote (single high-9)	’
“	left quote (double high-6)	“
”	right quote (double high-9)	”
„	double quote (double low-9)	„

相关页面

HTML DOM 参考手册：quotes 属性

Grid 属性

属性	描述	CSS
grid-columns	规定网格中每个列的宽度。	3
grid-rows	规定网格中每个列的高度。	3

CSS3 grid-columns 属性

实例

在 div 元素中部添加一个网格行，距右侧 200 像素处添加另一个，在余下空间的中间再添加一个：

```
div
{
grid-columns:50% * * 200px;
}
```

浏览器支持

目前没有浏览器支持 grid-columns。

定义和用法

grid-columns 属性网格中每列的宽度。

提示：使用网格系统对于打印设计师来说具有巨大的价值。现在相同的改变被应用到在线内容上。网格属性提供了在可伸缩网格中调整标题、文本和图片尺寸和位置的能力。

默认值：	**none**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.gridColumns="50% 200px"

语法

```
grid-columns: _length_|_%_|none|inherit;
```

值	描述
length	参考包含块的网格。
%	参考包含块的宽度。
none	
inherit	

CSS3 grid-rows 属性

实例

定义 100 像素的标题行，并按需添加多个额外的行，高度交替为 30 和 60 像素：

```
div
{
grid-rows:100px (30px 60px);
}
```

浏览器支持

目前没有浏览器支持 grid-rows。

定义和用法

grid-rows 属性规定网格中每行的高度。

提示：使用网格系统对于打印设计师来说具有巨大的价值。现在相同的改变被应用到在线内容上。网格属性提供了在可伸缩网格中调整标题、文本和图片尺寸和位置的能力。

默认值：	**none**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.gridRows="100px (30px 60px)"

语法

```
grid-rows: _length_|_%_|none|inherit;
```

值	描述
length	参考包含块的网格。
%	参考包含块的高度。
none	
inherit	

Hyperlink 属性

属性	描述	CSS
target	简写属性，设置target-name、target-new以及target-position属性。	3
target-name	规定在何处打开链接（链接的目标）。	3
target-new	规定目标链接在新窗口还是在已有窗口的新标签页中打开。	3
target-position	规定在何处放置新的目标链接。	3

CSS3 target 属性

实例

在新窗口中打开所有超链接，并在所有其他标签页/窗口上面放置新窗口：

```
a
{
target:new front;
}
```

浏览器支持

目前没有浏览器支持 target。

定义和用法

target 属性是一个简写属性，用于设置以下属性：

- target-name
- target-new
- target-position

默认值：	current window above
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.target="new front"

语法

```
target: _target-name_ _target-new_ _target-position_;
```

值	描述
target-name	规定在何处打开超链接（target destination）。
target-new	规定应该在新窗口或已有窗口的新标签页中打开超链接。
target-position	规定在何处放置新的目的地链接。

注释：target-new 和 target-position 值只有在 target-name 值创建新标签页或新窗口中有效。

CSS3 target-name 属性

实例

在新窗口中打开所有超链接：

```
a
{
target-name:new;
}
```

浏览器支持

目前没有浏览器支持 target-name。

定义和用法

target-name 属性规定在何处打开超链接 (target destination)。

默认值：	current
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.targetName="new"

语法

```
target-name: current|root|parent|new|modal|_name_;
```

值	描述
current	在链接所在的框架、标签页或窗口中打开超链接。
root	在当前标签页或窗口中超链接。
parent	在父框架中打开超链接。如果当前框架没有父框架，则将该值视作 root。
new	创建新的目的地（参阅 target-new）。
modal	在新的（暂时创建的）模态窗口中打开新窗口。
name	在已有框架、窗口或标签页中打开链接。如果 name 目的地不存在，则用该名称创建新的目的地。

CSS3 target-new 属性

实例

在新标签页而不是新窗口中打开超链接：

```
a
{
target-name:new;
target-new:tab;
}
```

浏览器支持

目前没有浏览器支持 target-new。

定义和用法

target-new 属性规定在新窗口还是新标签页或已有窗口中打开新的目的地链接。

注释：target-new 属性只有在 target-name 属性创建新标签页或新窗口时有效。

默认值：	**window**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.targetNew="tab"

语法

```
target-new: window|tab|none;
```

值	描述
window	在新窗口中打开超链接。
tab	在已有窗口的新标签页中打开超链接。
none	不创建新的目的地。

CSS3 target-position 属性

实例

在新窗口中打开超链接，并在所有其他标签页/窗口之前放置新的窗口：

```
a
{
target-name:new;
target-position:front;
}
```

浏览器支持

目前没有浏览器支持 target-position。

定义和用法

target-position 属性规定在何处放置新的目的地链接。

注释：target-position 属性只有在 target-name 属性创建新标签页或新窗口时有效。

默认值：	**above**
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.targetPosition="front"
---	---

语法

```
target-position: above|behind|front|back;
```

值	描述
above	在当前标签页/窗口之前放置新的目的地标签页/窗口。
behind	在当前标签页/窗口之后放置新的目的地标签页/窗口。
front	在所有其他标签页/窗口之前放置新的目的地标签页/窗口。
back	在所有其他标签页/窗口之后放置新的目的地标签页/窗口。

CSS 列表属性（List）

属性	描述	CSS
list-style	在一个声明中设置所有的列表属性。	1
list-style-image	将图象设置为列表项标记。	1
list-style-position	设置列表项标记的放置位置。	1
list-style-type	设置列表项标记的类型。	1
marker-offset	2	

CSS list-style 属性

实例

把图像设置为列表中的列表项目标记：

```
ul
  {
  list-style:square inside url('/i/arrow.gif');
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 list-style 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

list-style 简写属性在一个声明中设置所有的列表属性。

说明

该属性是一个简写属性，涵盖了所有其他列表样式属性。由于它应用到所有 display 为 list-item 的元素，所以在普通的 HTML 和 XHTML 中只能用于 li 元素，不过实际上它可以应用到任何元素，并由 list-item 元素继承。

可以按顺序设置如下属性：

- list-style-type
- list-style-position
- list-style-image

可以不设置其中的某个值，比如 "list-style:circle inside;" 也是允许的。未设置的属性会使用其默认值。

默认值：	disc outside none
继承性：	yes
版本：	CSS1
JavaScript 语法：	*object*.style.listStyle="decimal inside"

可能的值

值	描述
list-style-type	设置列表项标记的类型。参阅：list-style-type 中可能的值。
list-style-position	设置在何处放置列表项标记。参阅：list-style-position 中可能的值。
list-style-image	使用图像来替换列表项的标记。参阅：list-style-image 中可能的值。
inherit	规定应该从父元素继承 list-style 属性的值。

TIY 实例

在一个声明中定义所有的列表属性

本例演示将所有针对列表的属性设置于一个简写属性。

```
<html>
<head>
<style type="text/css">
ul
{
list-style: square inside url('/i/eg_arrow.gif')
}
</style>
</head>

<body>
<ul>
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ul>
</body>

</html>
```

相关页面

CSS 教程：CSS 列表

HTML DOM 参考手册：listStyle 属性

CSS list-style-image 属性

实例

把图像设置为列表中的项目标记：

```
ul
  {
  list-style-image:url("/i/arrow.gif");
  list-style-type:square;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 list-style-image 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

list-style-image 属性使用图像来替换列表项的标记。

说明

这个属性指定作为一个有序或无序列表项标志的图像。图像相对于列表项内容的放置位置通常使用 list-style-position 属性控制。

注释：请始终规定一个 "list-style-type" 属性以防图像不可用。

默认值：	none
继承性：	yes
版本：	CSS1
JavaScript 语法：	*object*.style.listStyleImage="url('/images/blueball.gif')"

可能的值

值	描述
URL	图像的路径。
none	默认。无图形被显示。
inherit	规定应该从父元素继承 list-style-image 属性的值。

TIY 实例

将图像作为列表项标记

本例演示如何将图像作为列表项标记。

```
<html>
<head>
<style type="text/css">
ul
{
list-style-image: url('/i/eg_arrow.gif')
}
</style>
</head>

<body>
<ul>
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ul>
</body>

</html>
```

相关页面

CSS 教程：CSS 列表

CSS 参考手册：CSS list-style 属性

HTML DOM 参考手册：listStyleImage 属性

CSS list-style-position 属性

实例

规定列表中列表项目标记的位置：

```
ul
  {
  list-style-position:inside;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 list-style-position 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

list-style-position 属性设置在何处放置列表项标记。

说明

该属性用于声明列表标志相对于列表项内容的位置。外部 (outside) 标志会放在离列表项边框边界一定距离处，不过这距离在 CSS 中未定义。内部 (inside) 标志处理为好像它们是插入在列表项内容最前面的行内元素一样。

默认值：	outside
继承性：	yes
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.listStylePosition="inside"
---	---

可能的值

值	描述
inside	列表项目标记放置在文本以内，且环绕文本根据标记对齐。
outside	默认值。保持标记位于文本的左侧。列表项目标记放置在文本以外，且环绕文本不根据标记对齐。
inherit	规定应该从父元素继承 list-style-position 属性的值。

TIY 实例

放置列表标记

本例演示在何处放置列表标记。

```
<html>
<head>
<style type="text/css">
ul.inside
{
list-style-position: inside
}

ul.outside
{
list-style-position: outside
}
</style>
</head>

<body>
<p>该列表的 list-style-position 的值是 "inside"：</p>
<ul class="inside">
<li>Earl Grey Tea - 一种黑颜色的茶</li>
<li>Jasmine Tea - 一种神奇的“全功能”茶</li>
<li>Honeybush Tea - 一种令人愉快的果味茶</li>
</ul>

<p>该列表的 list-style-position 的值是 "outside"：</p>
<ul class="outside">
<li>Earl Grey Tea - 一种黑颜色的茶</li>
<li>Jasmine Tea - 一种神奇的“全功能”茶</li>
<li>Honeybush Tea - 一种令人愉快的果味茶</li>
</ul>
</body>
</html>
```

相关页面

CSS 教程：CSS 列表

CSS 参考手册：CSS list-style 属性

HTML DOM 参考手册：listStylePosition 属性

CSS list-style-type 属性

实例

设置不同的列表样式：

```
ul.circle {list-style-type:circle;}
ul.square {list-style-type:square;}
ol.upper-roman {list-style-type:upper-roman;}
ol.lower-alpha {list-style-type:lower-alpha;}
```

（在页面底部可以找到更多实例）

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 list-style-type 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "decimal-leading-zero"、"lower-greek"、"lower-latin"、"upper-latin"、"armenian"、"georgian" 或 "inherit"。

定义和用法

list-style-type 属性设置列表项标记的类型。

默认值：	disc
继承性：	yes
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.listStyleType="square"
---	---

可能的值

CSS2 的值：

值	描述
none	无标记。
disc	默认。标记是实心圆。
circle	标记是空心圆。
square	标记是实心方块。
decimal	标记是数字。
decimal-leading-zero	0开头的数字标记。(01, 02, 03, 等。)
lower-roman	小写罗马数字(i, ii, iii, iv, v, 等。)
upper-roman	大写罗马数字(I, II, III, IV, V, 等。)
lower-alpha	小写英文字母The marker is lower-alpha (a, b, c, d, e, 等。)
upper-alpha	大写英文字母The marker is upper-alpha (A, B, C, D, E, 等。)
lower-greek	小写希腊字母(alpha, beta, gamma, 等。)
lower-latin	小写拉丁字母(a, b, c, d, e, 等。)
upper-latin	大写拉丁字母(A, B, C, D, E, 等。)
hebrew	传统的希伯来编号方式
armenian	传统的亚美尼亚编号方式
georgian	传统的乔治亚编号方式(an, ban, gan, 等。)
cjk-ideographic	简单的表意数字
hiragana	标记是：a, i, u, e, o, ka, ki, 等。（日文片假名）
katakana	标记是：A, I, U, E, O, KA, KI, 等。（日文片假名）
hiragana-iroha	标记是：i, ro, ha, ni, ho, he, to, 等。（日文片假名）
katakana-iroha	标记是：I, RO, HA, NI, HO, HE, TO, 等。（日文片假名）

CSS2.1 的值：

```
disc | circle | square | decimal | decimal-leading-zero |
lower-roman | upper-roman | lower-greek | lower-latin | upper-latin |
armenian | georgian | none | inherit
```

TIY 实例

在无序列表中的不同类型的列表标记

本例演示在CSS中不同类型的列表项标记。

```
<html>
<head>
<style type="text/css">
ul.disc {list-style-type: disc}
ul.circle {list-style-type: circle}
ul.square {list-style-type: square}
ul.none {list-style-type: none}
</style>
</head>

<body>
<ul class="disc">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="circle">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="square">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="none">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ul>
</body>

</html>
```

在有序列表中不同类型的列表项标记

本例演示在CSS中不同类型的列表项标记。

```
<html>
<head>
<style type="text/css">
ol.decimal {list-style-type: decimal}
ol.lroman {list-style-type: lower-roman}
ol.uroman {list-style-type: upper-roman}
ol.lalpha {list-style-type: lower-alpha}
ol.ualpha {list-style-type: upper-alpha}
</style>
</head>

<body>
<ol class="decimal">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ol>

<ol class="lroman">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ol>

<ol class="uroman">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ol>

<ol class="lalpha">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ol>

<ol class="ualpha">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ol>
</body>

</html>
```

所有的列表样式类型

本例演示在CSS中所有不同类型的列表项标记。

```
<html>
<head>
<style type="text/css">
ul.none {list-style-type: none}
ul.disc {list-style-type: disc}
ul.circle {list-style-type: circle}
ul.square {list-style-type: square}
ul.decimal {list-style-type: decimal}
ul.decimal-leading-zero {list-style-type: decimal-leading-zero}
ul.lower-roman {list-style-type: lower-roman}
ul.upper-roman {list-style-type: upper-roman}
ul.lower-alpha {list-style-type: lower-alpha}
ul.upper-alpha {list-style-type: upper-alpha}
ul.lower-greek {list-style-type: lower-greek}
ul.lower-latin {list-style-type: lower-latin}
ul.upper-latin {list-style-type: upper-latin}
ul.hebrew {list-style-type: hebrew}
ul.armenian {list-style-type: armenian}
```

```
ul.georgian {list-style-type: georgian}
ul.cjk-ideographic {list-style-type: cjk-ideographic}
ul.hiragana {list-style-type: hiragana}
ul.katakana {list-style-type: katakana}
ul.hiragana-iroha {list-style-type: hiragana-iroha}
ul.katakana-iroha {list-style-type: katakana-iroha}
</style>
</head>

<body>
<ul class="none">
<li>"none" 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="disc">
<li>Disc 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="circle">
<li>Circle 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="square">
<li>Square 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="decimal">
<li>Decimal 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="decimal-leading-zero">
<li>Decimal-leading-zero 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="lower-roman">
<li>Lower-roman 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="upper-roman">
<li>Upper-roman 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="lower-alpha">
<li>Lower-alpha 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="upper-alpha">
<li>Upper-alpha 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="lower-greek">
<li>Lower-greek 类型</li>
```

```
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="lower-latin">
<li>Lower-latin 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="upper-latin">
<li>Upper-latin 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="hebrew">
<li>Hebrew 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="armenian">
<li>Armenian 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="georgian">
<li>Georgian 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="cjk-ideographic">
<li>Cjk-ideographic 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="hiragana">
<li>Hiragana 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="katakana">
<li>Katakana 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="hiragana-iroha">
<li>Hiragana-iroha 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

<ul class="katakana-iroha">
<li>Katakana-iroha 类型</li>
<li>茶</li>
<li>可口可乐</li>
</ul>

</body>
</html>
```

相关页面

CSS 教程：CSS 列表

CSS 参考手册：CSS list-style 属性

HTML DOM 参考手册：listStyleType 属性

CSS 外边距属性（Margin）

属性	描述	CSS
margin	在一个声明中设置所有外边距属性。	1
margin-bottom	设置元素的下外边距。	1
margin-left	设置元素的左外边距。	1
margin-right	设置元素的右外边距。	1
margin-top	设置元素的上外边距。	1

CSS margin 属性

实例

设置 p 元素的 4 个外边距：

```
p
  {
  margin:2cm 4cm 3cm 4cm;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 margin 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

margin 简写属性在一个声明中设置所有外边距属性。该属性可以有 1 到 4 个值。

说明

这个简写属性设置一个元素所有外边距的宽度，或者设置各边上外边距的宽度。

块级元素的垂直相邻外边距会合并，而行内元素实际上不占上下外边距。行内元素的的左右外边距不会合并。同样地，浮动元素的外边距也不会合并。允许指定负的外边距值，不过使用时要小心。

注释：允许使用负值。

例子 1

```
margin:10px 5px 15px 20px;
```

- 上外边距是 10px
- 右外边距是 5px
- 下外边距是 15px

- 左外边距是 20px

例子 2

```
margin:10px 5px 15px;
```

- 上外边距是 10px
- 右外边距和左外边距是 5px
- 下外边距是 15px

例子 3

```
margin:10px 5px;
```

- 上外边距和下外边距是 10px
- 右外边距和左外边距是 5px

例子 4

```
margin:10px;
```

- 所有 4 个外边距都是 10px

默认值：	**0**
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.margin="10px 5px"

可能的值

值	描述
auto	浏览器计算外边距。
length	规定以具体单位计的外边距值，比如像素、厘米等。默认值是 0px。
%	规定基于父元素的宽度的百分比的外边距。
inherit	规定应该从父元素继承外边距。

TIY 实例

所有的边距属性在一个声明中

本例演示如何将所有的边距属性设置于一个声明中。

```
<html>
<head>
<style type="text/css">
p.margin {margin: 2cm 4cm 3cm 4cm}
</style>
</head>

<body>

<p>这个段落没有指定外边距。</p>

<p class="margin">这个段落带有指定的外边距。这个段落带有指定的外边距。这个段落带有指定的外边距。这个段落

<p>这个段落没有指定外边距。</p>

</body>

</html>
```

相关页面

CSS 教程：CSS 外边距

HTML DOM 参考手册：margin 属性

CSS margin-bottom 属性

实例

设置 p 元素的下外边距：

```
p
  {
  margin-bottom:2cm;
  }
```

（在页面底部可以找到更多实例）

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 margin-bottom 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

margin-bottom 属性设置元素的下外边距。

注释：允许使用负值。

默认值：	0
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.marginBottom="10px"

可能的值

值	描述
auto	浏览器计算下外边距。
length	规定以具体单位计的下外边距值，比如像素、厘米等。默认值是 0px。
%	规定基于父元素的宽度的百分比的下外边距。
inherit	规定应该从父元素继承下外边距。

TIY 实例

设置文本的下外边距 1

本例演示如何使用厘米值来设置文本的下外边距。

```
<html>
<head>
<style type="text/css">
p.bottommargin {margin-bottom: 2cm}
</style>
</head>

<body>
<p>这个段落没有指定外边距。</p>
<p class="bottommargin">这个段落带有指定的下外边距。</p>
<p>这个段落没有指定外边距。</p>
</body>

</html>
```

设置文本的下外边距 2

本例演示如何使用百分比值来设置文本的下外边距。

```
<html>
<head>
<style type="text/css">
p.bottommargin
{
margin-bottom: 25%
}
</style>
</head>
<body>

<p>This is a paragraph with no margin specifed</p>
<p class="bottommargin">This is a paragraph with a specified bottom margin</p>
<p>This is a paragraph with no margin specifed</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 外边距

HTML DOM 参考手册：marginBottom 属性

CSS margin-left 属性

实例

设置 p 元素的左外边距：

```
p
  {
  margin-left:2cm;
  }
```

（在页面底部可以找到更多实例）

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 margin-left 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

margin-left 属性设置元素的左外边距。

注释：允许使用负值。

默认值：	0
继承性：	no
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.marginLeft="10px"
---	---

可能的值

值	描述
auto	浏览器设置的左外边距。
length	定义固定的左外边距。默认值是0。
%	定义基于父对象总高度的百分比左外边距。
inherit	规定应该从父元素继承左外边距。

TIY 实例

设置文本的左外边距 1

本例演示如何使用厘米值来设置文本的左边距。

```
<html>
<head>
<style type="text/css">
p.leftmargin {margin-left: 2cm}
</style>
</head>

<body>
<p>这个段落没有指定外边距。</p>
<p class="leftmargin">这个段落带有指定的左外边距。</p>
</body>

</html>
```

设置文本的左外边距 2

本例演示如何使用百分比值来设置文本的左边距。

```
<html>
<head>
<style type="text/css">
p.leftmargin
{
margin-left: 25%
}
</style>
</head>
<body>

<p>This is a paragraph with no margin specified</p>
<p class="leftmargin">This is a paragraph with a specified left margin</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 外边距

HTML DOM 参考手册：marginLeft 属性

CSS margin-right 属性

实例

设置 p 元素的右外边距：

```
p
  {
  margin-right:2cm;
  }
```

（在页面底部可以找到更多实例）

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 margin-right 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

margin-right 属性设置元素的右外边距。

注释：允许使用负值。

默认值：	**0**
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.marginRight="10px"

可能的值

值	描述
auto	浏览器设置的右外边距。
length	定义固定的右外边距。默认值是 0。
%	定义基于父对象总高度的百分比右外边距。
inherit	规定应该从父元素继承右外边距。

TIY 实例

设置文本的右外边距 1

本例演示如何使用厘米值来设置文本的右外边距。

```
<html>
<head>
<style type="text/css">
p.rightmargin {margin-right: 8cm}
</style>
</head>

<body>
<p>这个段落没有指定外边距。</p>
<p class="rightmargin">这个段落带有指定的右外边距。这个段落带有指定的右外边距。这个段落带有指定的右外边
</body>

</html>
```

设置文本的右外边距 2

本例演示如何使用百分比值来设置文本的右外边距。

```
<html>
<head>
<style type="text/css">
p.rightmargin
{
margin-right:25%
}
</style>
</head>
<body>

<p style="text-align:right">This is a right aligned paragraph with no margin specified</p
<p class="rightmargin" style="text-align:right">This is a right aligned paragraph with a

</body>
</html>
```

相关页面

CSS 教程：CSS 外边距

HTML DOM 参考手册：marginRight 属性

CSS margin-top 属性

实例

设置 p 元素的上外边距：

```
p
  {
  margin-top:2cm;
  }
```

（在页面底部可以找到更多实例）

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 margin-top 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

margin-top 属性设置元素的上外边距。

注释：允许使用负值。

默认值：	0
继承性：	no
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.marginTop="10px"
---	---

可能的值

值	描述
auto	浏览器设置的上外边距。
length	定义固定的上外边距。默认值是 0。
%	定义基于父对象总高度的百分比上外边距。
inherit	规定应该从父元素继承上外边距。

TIY 实例

设置文本的上外边距 1

本例演示如何使用厘米值来设置文本的顶边距。

```
<html>
<head>
<style type="text/css">
p.topmargin {margin-top: 5cm}
</style>
</head>

<body>
<p>这个段落没有指定外边距。</p>
<p class="topmargin">这个段落带有指定的上外边距。</p>
</body>
</html>
```

设置文本的上外边距 2

本例演示如何使用百分比值来设置文本的顶边距。

```
<html>
<head>
<style type="text/css">
p.topmargin
{
margin-top: 25%
}
</style>
</head>
<body>

<p>This is a paragraph with no margin specified</p>
<p class="topmargin">This is a paragraph with a specified top margin</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 外边距

HTML DOM 参考手册：marginTop 属性

Marquee 属性

属性	描述	CSS
marquee-direction	设置移动内容的方向。	3
marquee-play-count	设置内容移动多少次。	3
marquee-speed	设置内容滚动得多快。	3
marquee-style	设置移动内容的样式。	3

多列属性（Multi-column）

属性	描述	CSS
column-count	规定元素应该被分隔的列数。	3
column-fill	规定如何填充列。	3
column-gap	规定列之间的间隔。	3
column-rule	设置所有 column-rule-* 属性的简写属性。	3
column-rule-color	规定列之间规则的颜色。	3
column-rule-style	规定列之间规则的样式。	3
column-rule-width	规定列之间规则的宽度。	3
column-span	规定元素应该横跨的列数。	3
column-width	规定列的宽度。	3
columns	规定设置 column-width 和 column-count 的简写属性。	3

CSS3 column-count 属性

实例

将 div 元素中的文本分为三列：

```
div
{
-moz-column-count:3; /* Firefox */
-webkit-column-count:3; /* Safari 和 Chrome */
column-count:3;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10 和 Opera 支持 column-count 属性。

Firefox 支持替代的 -moz-column-count 属性。

Safari 和 Chrome 支持替代的 -webkit-column-count 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 column-count 属性。

定义和用法

column-count 属性规定元素应该被划分的列数。

默认值：	auto
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.columnCount=3

语法

```
column-count: _number_|auto;
```

值	描述	测试
number	元素内容将被划分的最佳列数。	测试
auto	由其他属性决定列数，比如 "column-width"。	测试

亲自试一试 - 实例

Column-count

把 div 元素中的文本划分为三列。

```
<!DOCTYPE html>
<html>
<head>
<style>
.newspaper
{
-moz-column-count:3; /* Firefox */
-webkit-column-count:3; /* Safari and Chrome */
column-count:3;
}
</style>
</head>
<body>

<p><b>注释：</b>Internet Explorer 不支持 column-count 属性。</p>

<div class="newspaper">
人民网北京2月24日电 (记者 刘阳)国家发展改革委近日发出通知，决定自2月25日零时起将汽、柴油价格每吨分别提高3

此次国内成品油价格调整幅度，是按照现行国内成品油价格形成机制，根据国际市场油价变化情况确定的。去年11月16日

通知指出，这次成品油调价后，国家将按照已建立的补贴机制，继续对种粮农民、渔业（含远洋渔业）、林业、城市公交、

通知要求，中石油、中石化、中海油三大公司要组织好成品油生产和调运，保持合理库存，加强综合协调和应急调度，保障
</div>

</body>
</html>
```

相关页面

CSS3 教程：CSS3 多列

CSS3 column-fill 属性

实例

规定如何对列进行填充：

```
div
{
column-fill:auto;
}
```

浏览器支持

主流浏览器都不支持 column-fill 属性。

定义和用法

IE	Firefox	Chrome	Safari	Opera

column-fill 属性规定如何填充列（是否进行协调）。

默认值：	balance
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.columnFill="auto"
---	---

语法

```
column-fill: balance|auto;
```

值	描述
balance	对列进行协调。浏览器应对列长度的差异进行最小化处理。
auto	按顺序对列进行填充，列长度会各有不同。

相关页面

CSS3 教程：CSS3 多列

CSS3 column-gap 属性

实例

规定列间的间隔为 40 像素：

```
div
{
-moz-column-gap:40px; /* Firefox */
-webkit-column-gap:40px; /* Safari 和 Chrome */
column-gap:40px;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10 和 Opera 支持 column-gap 属性。

Firefox 支持替代的 -moz-column-gap 属性。

Safari 和 Chrome 支持替代的 -webkit-column-gap 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 column-gap 属性。

定义和用法

column-gap 属性规定列之间的间隔。

注释：如果列之间设置了 column-rule，它会在间隔中间显示。

默认值：	normal
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.columnGap="40px"

语法

```
column-gap: _length_|normal;
```

值	描述	测试
length	把列间的间隔设置为指定的长度。	测试
normal	规定列间间隔为一个常规的间隔。W3C 建议的值是 1em。	测试

亲自试一试 - 实例

Column-gap

将 div 元素中的文本分为三列，并规定列间 30 像素的间隔。

```
<!DOCTYPE html>
<html>
<head>
<style>
.newspaper
{
-moz-column-count:3; /* Firefox */
-webkit-column-count:3; /* Safari and Chrome */
column-count:3;

-moz-column-gap:30px; /* Firefox */
-webkit-column-gap:30px; /* Safari and Chrome */
column-gap:30px;
}
</style>
</head>
<body>

<p><b>注释：</b>Internet Explorer 不支持 column-count 属性。</p>

<div class="newspaper">
人民网北京2月24日电 (记者 刘阳)国家发展改革委近日发出通知，决定自2月25日零时起将汽、柴油价格每吨分别提高3

此次国内成品油价格调整幅度，是按照现行国内成品油价格形成机制，根据国际市场油价变化情况确定的。去年11月16日

通知指出，这次成品油调价后，国家将按照已建立的补贴机制，继续对种粮农民、渔业（含远洋渔业）、林业、城市公交、

通知要求，中石油、中石化、中海油三大公司要组织好成品油生产和调运，保持合理库存，加强综合协调和应急调度，保障
</div>

</body>
</html>
```

相关页面

CSS3 教程：CSS3 多列

CSS3 column-rule 属性

实例

规定列之间的宽度、样式和颜色规则：

```
div
{
-moz-column-rule:3px outset #ff00ff; /* Firefox */
-webkit-column-rule:3px outset #ff00ff; /* Safari 和 Chrome */
column-rule:3px outset #ff00ff;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10 和 Opera 支持 column-rule 属性。

Firefox 支持替代的 -moz-column-rule 属性。

Safari 和 Chrome 支持替代的 -webkit-column-rule 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 column-rule 属性。

定义和用法

column-rule 属性是一个简写属性，用于设置所有 column-rule-* 属性。

column-rule 属性设置列只觉得宽度、样式和颜色规则。

默认值：	**medium none black**
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.columnRule="3px outset #ff00ff"
---	---

语法

```
column-rule: _column-rule-width_ _column-rule-style_ _column-rule-color_;
```

值	描述
column-rule-width	设置列之间的宽度规则。
column-rule-style	设置列之间的样式规则。
column-rule-color	设置列之间的颜色规则。

亲自试一试 - 实例

Column-rule

规定列之间的宽度、样式和颜色。

```
<!DOCTYPE html>
<html>
<head>
<style>
.newspaper
{
-moz-column-count:3; /* Firefox */
-webkit-column-count:3; /* Safari and Chrome */
column-count:3;

-moz-column-gap:40px; /* Firefox */
-webkit-column-gap:40px; /* Safari and Chrome */
column-gap:40px;

-moz-column-rule:4px outset #ff0000; /* Firefox */
-webkit-column-rule:4px outset #ff0000; /* Safari and Chrome */
column-rule:4px outset #ff0000;
}
</style>
</head>
<body>

<p><b>注释：</b>Internet Explorer 不支持 column-count 属性。</p>

<div class="newspaper">
人民网北京2月24日电 (记者 刘阳)国家发展改革委近日发出通知，决定自2月25日零时起将汽、柴油价格每吨分别提高3

此次国内成品油价格调整幅度，是按照现行国内成品油价格形成机制，根据国际市场油价变化情况确定的。去年11月16日

通知指出，这次成品油调价后，国家将按照已建立的补贴机制，继续对种粮农民、渔业（含远洋渔业）、林业、城市公交、

通知要求，中石油、中石化、中海油三大公司要组织好成品油生产和调运，保持合理库存，加强综合协调和应急调度，保障
</div>

</body>
</html>
```

相关页面

CSS3 教程：CSS3 多列

CSS3 column-rule-color 属性

实例

设置列之间的颜色规则：

```
div
{
-moz-column-rule-color:#ff0000; /* Firefox */
-webkit-column-rule-color:#ff0000; /* Safari 和 Chrome */
column-rule-color:#ff0000;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10 和 Opera 支持 column-rule-color 属性。

Firefox 支持替代的 -moz-column-rule-color 属性。

Safari 和 Chrome 支持替代的 -webkit-column-rule-color 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 column-rule-color 属性。

定义和用法

column-rule-color 属性规定列之间的颜色规则。

默认值：	black
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.columnRuleColor="#ff00ff"

语法

```
column-rule-color: _color_;
```

值	描述
color	规定颜色规则。请参阅 CSS 颜色值。

相关页面

CSS3 教程：CSS3 多列

CSS3 column-rule-style 属性

实例

设置列之间的颜色规则：

```
div
{
-moz-column-rule-style:dotted; /* Firefox */
-webkit-column-rule-style:dotted; /* Safari 和 Chrome */
column-rule-style:dotted;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10 和 Opera 支持 column-rule-style 属性。

Firefox 支持替代的 -moz-column-rule-style 属性。

Safari 和 Chrome 支持替代的 -webkit-column-rule-style 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 column-rule-style 属性。

定义和用法

column-rule-style 属性规定列之间的样式规则。

默认值：	none
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.columnRuleStyle="dotted"
---	---

语法

```
column-rule-style: none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset;
```

值	描述	测试
none	定义没有规则。	测试
hidden	定义隐藏规则。	测试
dotted	定义点状规则。	测试
dashed	定义虚线规则。	测试
solid	定义实线规则。	测试
double	定义双线规则。	测试
groove	定义 3D grooved 规则。该效果取决于宽度和颜色值。	测试
ridge	定义 3D ridged 规则。该效果取决于宽度和颜色值。	测试
inset	定义 3D inset 规则。该效果取决于宽度和颜色值。	测试
outset	定义 3D outset 规则。该效果取决于宽度和颜色值。	测试

相关页面

CSS3 教程：CSS3 多列

CSS3 column-rule-width 属性

实例

设置列之间的颜色规则：

```
div
{
-moz-column-rule-width:1px; /* Firefox */
-webkit-column-rule-width:1px; /* Safari 和 Chrome */
column-rule-width:1px;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10 和 Opera 支持 column-rule-width 属性。

Firefox 支持替代的 -moz-column-rule-width 属性。

Safari 和 Chrome 支持替代的 -webkit-column-rule-width 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 column-rule-width 属性。

定义和用法

column-rule-width 属性规定列之间的宽度规则。

默认值：	medium
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.columnRuleWidth="thin"

语法

```
column-rule-width: thin|medium|thick|_length_;
```

值	描述	测试
thin	定义纤细规则。	测试
medium	定义中等规则。	测试
thick	定义宽厚规则。	测试
length	规定规则的宽度。	测试

相关页面

CSS3 教程：CSS3 多列

CSS3 column-span 属性

实例

使 h2 元素横跨所有列：

```
h2
{
-webkit-column-span:all; /* Chrome */
column-span:all;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10 和 Opera 支持 column-span 属性。

Safari 和 Chrome 支持替代的 -webkit-column-span 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 column-span 属性。

定义和用法

column-span 属性规定元素应横跨多少列。

默认值：	1
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.columnSpan="all"

语法

```
column-span: 1|all;
```

值	描述	测试
1	元素应横跨一列。	测试
all	元素应横跨所有列。	测试

相关页面

CSS3 教程：CSS3 多列

CSS3 column-width 属性

实例

规定列的宽度：

```
div
{
column-width:100px;
-moz-column-width:100px; /* Firefox */
-webkit-column-width:100px; /* Safari 和 Chrome */
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10 和 Opera 支持 column-width 属性。

Firefox 支持替代的 -moz-column-width 属性。

Safari 和 Chrome 支持替代的 -webkit-column-width 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 column-width 属性。

定义和用法

column-width 属性规定列的宽度。

默认值：	auto
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.columnWidth="100px"

语法

```
column-width: auto|_length_;
```

值	描述	测试
auto	由浏览器决定列宽。	测试
length	规定列的宽度。	测试

相关页面

CSS3 教程：CSS3 多列

CSS3 columns 属性

实例

规定列的宽度和列数：

```
div
{
columns:100px 3;
-moz-columns:100px 3; /* Firefox */
-webkit-columns:100px 3; /* Safari 和 Chrome */
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10 和 Opera 支持 column 属性。

Firefox 支持替代的 -moz-column 属性。

Safari 和 Chrome 支持替代的 -webkit-column 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 column 属性。

定义和用法

columns 属性是一个简写属性，用于设置列宽和列数。

默认值：	auto auto
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.columns="100px 3"

语法

```
columns: _column-width_ _column-count_;
```

值	描述
column-width	列的宽度。
column-count	列数。

相关页面

CSS3 教程：CSS3 多列

CSS 内边距属性（Padding）

属性	描述	CSS
padding	在一个声明中设置所有内边距属性。	1
padding-bottom	设置元素的下内边距。	1
padding-left	设置元素的左内边距。	1
padding-right	设置元素的右内边距。	1
padding-top	设置元素的上内边距。	1

CSS padding 属性

实例

设置 p 元素的 4 个内边距：

```
p
  {
  padding:2cm 4cm 3cm 4cm;
  }
```

浏览器支持

所有浏览器都支持 padding 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

padding 简写属性在一个声明中设置所有内边距属性。

说明

这个简写属性设置元素所有内边距的宽度，或者设置各边上内边距的宽度。行内非替换元素上设置的内边距不会影响行高计算；因此，如果一个元素既有内边距又有背景，从视觉上看可能会延伸到其他行，有可能还会与其他内容重叠。元素的背景会延伸穿过内边距。不允许指定负边距值。

注释：不允许使用负值。

例子 1

```
padding:10px 5px 15px 20px;
```

- 上内边距是 10px
- 右内边距是 5px
- 下内边距是 15px
- 左内边距是 20px

例子 2

```
padding:10px 5px 15px;
```

- 上内边距是 10px
- 右内边距和左内边距是 5px
- 下内边距是 15px

例子 3

```
padding:10px 5px;
```

- 上内边距和下内边距是 10px
- 右内边距和左内边距是 5px

例子 4

```
padding:10px;
```

- 所有 4 个内边距都是 10px

默认值：	**0**
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.padding="10px 5px"

可能的值

值	描述
auto	浏览器计算内边距。
length	规定以具体单位计的内边距值，比如像素、厘米等。默认值是 0px。
%	规定基于父元素的宽度的百分比的内边距。
inherit	规定应该从父元素继承内边距。

TIY 实例

所有内边距属性在一个声明中

本例演示使用简写属性将所有的内边距属性设置于一个声明中，可以有一到四个值。

```
<html>
<head>
<style type="text/css">
td.test1 {padding: 1.5cm}
td.test2 {padding: 0.5cm 2.5cm}
</style>
</head>

<body>
<table border="1">
<tr>
<td class="test1">
这个表格单元的每个边拥有相等的内边距。
</td>
</tr>
</table>
<br />
<table border="1">
<tr>
<td class="test2">
这个表格单元的上和下内边距是 0.5cm，左和右内边距是 2.5cm。
</td>
</tr>
</table>
</body>

</html>
```

相关页面

CSS 教程：CSS 内边距

HTML DOM 参考手册：padding 属性

CSS padding-bottom 属性

实例

设置 p 元素的下内边距：

```
p
  {
  padding-bottom:2cm;
  }
```

（在页面底部可以找到更多实例）

浏览器支持

所有主流浏览器都支持 padding-bottom 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

padding-bottom 属性设置元素的下内边距（底部空白）。

说明

该属性设置元素下内边距的宽度。行内非替换元素上设置的下内边距不会影响行高计算，因此，如果一个元素既有内边距又有背景，从视觉上看可能延伸到其他行，有可能还会与其他内容重叠。不允许指定负内边距值。

注释：不允许使用负值。

默认值：	**0**
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.paddingBottom="10px"

可能的值

值	描述
length	规定以具体单位计的固定的下内边距值，比如像素、厘米等。默认值是 0px。
%	定义基于父元素宽度的百分比下内边距。此值不会如预期地那样工作于所有的浏览器中。
inherit	规定应该从父元素继承下内边距。

TIY 实例

设置下内边距 1

本例演示如何使用厘米值来设置单元格的下内边距。

```
<html>
<head>
<style type="text/css">
td {padding-bottom: 2cm}
</style>
</head>

<body>
<table border="1">
<tr>
<td>
这个表格单元拥有下内边距。
</td>
</tr>
</table>
</body>

</html>
```

设置下内边距 2

本例演示如何使用百分比值来设置单元格的下内边距。

```
<html>
<head>
<style type="text/css">
td
{
padding-bottom: 10%
}
</style>
</head>
<body>

<table border="1">
<tr>
<td>
这个表格单元拥有下内边距。
</td>
</tr>
</table>

</body>
</html>
```

相关页面

CSS 教程：CSS 内边距

HTML DOM 参考手册：paddingBottom 属性

CSS padding-left 属性

实例

设置 p 元素的左内边距：

```
p
  {
  padding-left:2cm;
  }
```

（在页面底部可以找到更多实例）

浏览器支持

所有主流浏览器都支持 padding-left 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

padding-left 属性设置元素左内边距（空白）。

说明

该属性设置元素左内边距的宽度。行内非替换元素上设置的左内边距仅在元素所生成的第一个行内框的左边出现.

注释：不允许使用负值。

默认值：	**0**
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.paddingLeft="10px"

可能的值

值	描述
length	规定以具体单位计的固定的左内边距值，比如像素、厘米等。默认值是 0px。
%	定义基于父元素宽度的百分比左内边距。此值不会如预期地那样工作于所有的浏览器中。
inherit	规定应该从父元素继承左内边距。

TIY 实例

设置左内边距 1

本例演示如何使用厘米值来设置单元格的左内边距。

```
<html>
<head>
<style type="text/css">
td {padding-left: 2cm}
</style>
</head>

<body>
<table border="1">
<tr>
<td>
这个表格单元拥有左内边距。
</td>
</tr>
</table>
</body>

</html>
```

设置左内边距 2

本例演示如何使用百分比值来设置单元格的左内边距。

```
<html>
<head>
<style type="text/css">
td
{
padding-left: 10%
}
</style>
</head>
<body>

<table border="1">
<tr>
<td>
这个表格单元拥有左内边距。
</td>
</tr>
</table>

</body>
</html>
```

相关页面

CSS 教程：CSS 内边距

HTML DOM 参考手册：paddingLeft 属性

CSS padding-right 属性

实例

设置 p 元素的右内边距：

```
p
  {
  padding-right:2cm;
  }
```

（在页面底部可以找到更多实例）

浏览器支持

所有主流浏览器都支持 padding-right 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

padding-right 属性设置元素右内边距（空白）。

注释：不允许使用负值。

说明

该属性设置元素右内边距的宽度。行内非替换元素上设置的右内边距仅在元素所生成的第一个行内框的右边出现.

默认值：	**0**
继承性：	no
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.paddingRight="10px"
---	---

可能的值

值	描述
length	规定以具体单位计的固定的右内边距值，比如像素、厘米等。默认值是 0px。
%	定义基于父元素宽度的百分比右内边距。此值不会如预期地那样工作于所有的浏览器中。
inherit	规定应该从父元素继承右内边距。

TIY 实例

设置右内边距 1

本例演示如何使用厘米值来设置单元格的右内边距。

```
<html>
<head>
<style type="text/css">
td {padding-right: 5cm}
</style>
</head>

<body>
<table border="1">
<tr>
<td>
这个表格单元拥有右内边距。
</td>
</tr>
</table>
</body>

</html>
```

设置右内边距 2

本例演示如何使用百分比值来设置单元格的右内边距。

```
<html>
<head>
<style type="text/css">
td
{
padding-right: 10%
}
</style>
</head>
<body>

<table border="1">
<tr>
<td>
这个表格单元拥有右内边距。
</td>
</tr>
</table>

</body>
</html>
```

相关页面

CSS 教程：CSS 内边距

HTML DOM 参考手册：paddingRight 属性

CSS padding-top 属性

实例

设置 p 元素的上内边距：

```
p
  {
  padding-top:2cm;
  }
```

（在页面底部可以找到更多实例）

浏览器支持

所有主流浏览器都支持 padding-top 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

padding-top 属性设置元素的上内边距（空间）。

说明

该属性设置元素上内边距的宽度。行内非替换元素上设置的上内边距不会影响行高计算，因此，如果一个元素既有内边距又有背景，从视觉上看可能延伸到其他行，有可能还会与其他内容重叠。不允许指定负内边距值。

注释：不允许使用负值。

默认值：	**0**
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.paddingTop="10px"

可能的值

值	描述
length	规定以具体单位计的固定的上内边距值，比如像素、厘米等。默认值是 0px。
%	定义基于父元素宽度的百分比上内边距。此值不会如预期的那样工作于所有的浏览器中。
inherit	规定应该从父元素继承上内边距。

TIY 实例

设置上内边距 1

本例演示如何使用厘米值来设置单元格的上内边距。

```
<html>
<head>
<style type="text/css">
td {padding-top: 2cm}
</style>
</head>

<body>
<table border="1">
<tr>
<td>
这个表格单元拥有上内边距。
</td>
</tr>
</table>
</body>

</html>
```

设置上内边距 2

本例演示如何使用百分比值来设置单元格的上内边距。

```
<html>
<head>
<style type="text/css">
td
{
padding-top: 10%
}
</style>
</head>
<body>

<table border="1">
<tr>
<td>
这个表格单元拥有上内边距。
</td>
</tr>
</table>

</body>
</html>
```

相关页面

CSS 教程：CSS 内边距

HTML DOM 参考手册：paddingTop 属性

Paged Media 属性

属性	描述	CSS
fit	示意如何对width和height属性均不是auto的被替换元素进行缩放。	3
fit-position	定义盒内对象的对齐方式。	3
image-orientation	规定用户代理应用于图像的顺时针方向旋转。	3
page	规定元素应该被显示的页面特定类型。	3
size	规定页面内容包含框的尺寸和方向。	3

CSS 定位属性（Positioning）

属性	描述	CSS
bottom	设置定位元素下外边距边界与其包含块下边界之间的偏移。	2
clear	规定元素的哪一侧不允许其他浮动元素。	1
clip	剪裁绝对定位元素。	2
cursor	规定要显示的光标的类型（形状）。	2
display	规定元素应该生成的框的类型。	1
float	规定框是否应该浮动。	1
left	设置定位元素左外边距边界与其包含块左边界之间的偏移。	2
overflow	规定当内容溢出元素框时发生的事情。	2
position	规定元素的定位类型。	2
right	设置定位元素右外边距边界与其包含块右边界之间的偏移。	2
top	设置定位元素的上外边距边界与其包含块上边界之间的偏移。	2
vertical-align	设置元素的垂直对齐方式。	1
visibility	规定元素是否可见。	2
z-index	设置元素的堆叠顺序。	2

CSS bottom 属性

实例

把图像的底边缘设置在其包含元素底边缘之上 5 像素高的位置：

```
img
  {
  position:absolute;
  bottom:5px;
  }
```

浏览器支持

所有主流浏览器都支持 bottom 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

bottom 属性规定元素的底部边缘。该属性定义了定位元素下外边距边界与其包含块下边界之间的偏移。

注释：如果 "position" 属性的值为 "static"，那么设置 "bottom" 属性不会产生任何效果。

说明

对于 static 元素，为 auto；对于长度值，则为相应的绝对长度；对于百分比数值，为指定值；否则为 auto。

对于相对定义元素，如果 bottom 和 top 都是 auto，其计算值则都是 0；如果其中之一为 auto，则取另一个值的相反数；如果二者都不是 auto，bottom 将取 top 值的相反数。

默认值：	auto
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.bottom="50px"

可能的值

值	描述
auto	默认值。通过浏览器计算底部的位置。
%	设置以包含元素的百分比计的底边位置。可使用负值。
length	使用 px、cm 等单位设置元素的底边位置。可使用负值。
inherit	规定应该从父元素继承 bottom 属性的值。

TIY 实例

使用像素值设置图像的底部边缘

本例演示如何使用像素值设置图像的底部边缘。

```
<html>
<head>
<style type="text/css">
img
{
position:absolute;
bottom:0px
}
</style>
</head>
<body>

<h1>这是标题</h1>
<img class="normal" src="/i/eg_smile.gif" />
<p>一些文本。一些文本。一些文本。一些文本。一些文本。一些文本。</p>

</body>
</html>
```

使用百分比设置图像的底部边缘

本例演示如何使用百分比值设置图像的底部边缘。

```
<html>
<head>
<style type="text/css">
img
{
position:absolute;
bottom:5%
}
</style>
</head>
<body>

<h1>这是标题</h1>
<img class="normal" src="/i/eg_smile.gif" />
<p>一些文本。一些文本。一些文本。一些文本。一些文本。一些文本。</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：bottom 属性

CSS clear 属性

实例

图像的左侧和右侧均不允许出现浮动元素：

```
img
  {
  float:left;
  clear:both;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 clear 属性。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

clear 属性规定元素的哪一侧不允许其他浮动元素。

说明

clear 属性定义了元素的哪边上不允许出现浮动元素。在 CSS1 和 CSS2 中，这是通过自动为清除元素（即设置了 clear 属性的元素）增加上外边距实现的。在 CSS2.1 中，会在元素上外边距之上增加清除空间，而外边距本身并不改变。不论哪一种改变，最终结果都一样，如果声明为左边或右边清除，会使元素的上外边框边界刚好在该边上浮动元素的下外边距边界之下。

默认值：	none
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.clear="left"

可能的值

值	描述
left	在左侧不允许浮动元素。
right	在右侧不允许浮动元素。
both	在左右两侧均不允许浮动元素。
none	默认值。允许浮动元素出现在两侧。
inherit	规定应该从父元素继承 clear 属性的值。

TIY 实例

清除元素的侧面

本例演示如何使用清除元素侧面的浮动元素。

```
<html>

<head>
<style type="text/css">
img
  {
  float:left;
  clear:both;
  }
</style>
</head>

<body>
<img src="/i/eg_smile.gif" />
<img src="/i/eg_smile.gif" />
</body>

</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：clear 属性

CSS clip 属性

实例

剪裁图像：

```
img
  {
  position:absolute;
  clip:rect(0px,60px,200px,0px);
  }
```

浏览器支持

所有主流浏览器都支持 clip 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

clip 属性剪裁绝对定位元素。

当一幅图像的尺寸大于包含它的元素时会发生什么呢？"clip" 属性允许您规定一个元素的可见尺寸，这样此元素就会被修剪并显示为这个形状。

说明

这个属性用于定义一个剪裁矩形。对于一个绝对定义元素，在这个矩形内的内容才可见。出了这个剪裁区域的内容会根据 overflow 的值来处理。剪裁区域可能比元素的内容区大，也可能比内容区小。

默认值：	auto
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.clip="rect(0px,50px,50px,0px)"

可能的值

值	描述
shape	设置元素的形状。唯一合法的形状值是：rect (*top*, *right*, *bottom*, *left*)
auto	默认值。不应用任何剪裁。
inherit	规定应该从父元素继承 clip 属性的值。

TIY 实例

设置元素的形状

本例演示如何设置元素的形状。此元素被剪入形状中，然后显示出来。

```
<html>
<head>
<style type="text/css">
img
{
position:absolute;
clip:rect(0px 50px 200px 0px)
}
</style>
</head>

<body>
<p>clip 属性剪切了一幅图像：</p>
<p><img border="0" src="/i/eg_bookasp.gif" width="120" height="151"></p>
</body>

</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：clip 属性

CSS cursor 属性

实例

一些不同的光标：

```
span.crosshair {cursor:crosshair;}
span.help {cursor:help;}
span.wait {cursor:wait;}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 cursor 属性。

注释：Opera 9.3 和 Safari 3 不支持 *url* 值。

注释：任何版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

cursor 属性规定要显示的光标的类型（形状）。

该属性定义了鼠标指针放在一个元素边界范围内时所用的光标形状（不过 CSS2.1 没有定义由哪个边界确定这个范围）。

默认值：	auto
继承性：	yes
---	---
版本：	CSS2
---	---
JavaScript 语法：	*object*.style.cursor="crosshair"
---	---

可能的值

值	描述
url	

需使用的自定义光标的 URL。

注释：请在此列表的末端始终定义一种普通的光标，以防没有由 URL 定义的可用光标。

| | default | 默认光标（通常是一个箭头） | | auto | 默认。浏览器设置的光标。 | | crosshair | 光标呈现为十字线。 | | pointer | 光标呈现为指示链接的指针（一只手） | | move | 此光标指示某对象可被移动。 | | e-resize | 此光标指示矩形框的边缘可被向右（东）移动。 | | ne-resize | 此光标指示矩形框的边缘可被向上及向右移动（北/东）。 | | nw-resize | 此光标指示矩形框的边缘可被向上及向左移动（北/西）。 | | n-resize | 此光标指示矩形框的边缘可被向上（北）移动。 | | se-resize | 此光标指示矩形框的边缘可被向下及向右移动（南/东）。 | | sw-resize | 此光标指示矩形框的边缘可被向下及向左移动（南/西）。 | | s-resize | 此光标指示矩形框的边缘可被向下移动（南）。 | | w-resize | 此光标指示矩形框的边缘可被向左移动（西）。 | | text | 此光标指示文本。 | | wait | 此光标指示程序正忙（通常是一只表或沙漏）。 | | help | 此光标指示可用的帮助（通常是一个问号或一个气球）。 |

TIY 实例

改变光标

本例演示如何改变光标。

```
<html>

<body>
<p>请把鼠标移动到单词上，可以看到鼠标指针发生变化：</p>
<span style="cursor:auto">
Auto</span><br />
<span style="cursor:crosshair">
Crosshair</span><br />
<span style="cursor:default">
Default</span><br />
<span style="cursor:pointer">
Pointer</span><br />
<span style="cursor:move">
Move</span><br />
<span style="cursor:e-resize">
e-resize</span><br />
<span style="cursor:ne-resize">
ne-resize</span><br />
<span style="cursor:nw-resize">
nw-resize</span><br />
<span style="cursor:n-resize">
n-resize</span><br />
<span style="cursor:se-resize">
se-resize</span><br />
<span style="cursor:sw-resize">
sw-resize</span><br />
<span style="cursor:s-resize">
s-resize</span><br />
<span style="cursor:w-resize">
w-resize</span><br />
<span style="cursor:text">
text</span><br />
<span style="cursor:wait">
wait</span><br />
<span style="cursor:help">
help</span>
</body>

</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：cursor 属性

CSS display 属性

实例

使段落生出行内框：

```
p.inline
  {
  display:inline;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 display 属性。

注释：如果规定了 !DOCTYPE，则 Internet Explorer 8 （以及更高版本）支持属性值 "inline-table"、"run-in"、"table"、"table-caption"、"table-cell"、"table-column"、"table-column-group"、"table-row"、"table-row-group"、以及 "inherit"。

定义和用法

display 属性规定元素应该生成的框的类型。

说明

这个属性用于定义建立布局时元素生成的显示框类型。对于 HTML 等文档类型，如果使用 display 不谨慎会很危险，因为可能违反 HTML 中已经定义的显示层次结构。对于 XML，由于 XML 没有内置的这种层次结构，所有 display 是绝对必要的。

注释：CSS2 中有值 compact 和 marker，不过由于缺乏广泛的支持，已经从 CSS2.1 中去除了。

默认值：	**inline**
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.display="inline"

可能的值

值	描述
none	此元素不会被显示。
block	此元素将显示为块级元素，此元素前后会带有换行符。
inline	默认。此元素会被显示为内联元素，元素前后没有换行符。
inline-block	行内块元素。（CSS2.1 新增的值）
list-item	此元素会作为列表显示。
run-in	此元素会根据上下文作为块级元素或内联元素显示。
compact	CSS 中有值 compact，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。
marker	CSS 中有值 marker，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。
table	此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。
inline-table	此元素会作为内联表格来显示（类似 <table>），表格前后没有换行符。
table-row-group	此元素会作为一个或多个行的分组来显示（类似 <tbody>）。
table-header-group	此元素会作为一个或多个行的分组来显示（类似 <thead>）。
table-footer-group	此元素会作为一个或多个行的分组来显示（类似 <tfoot>）。
table-row	此元素会作为一个表格行显示（类似 <tr>）。
table-column-group	此元素会作为一个或多个列的分组来显示（类似 <colgroup>）。
table-column	此元素会作为一个单元格列显示（类似 <col>）
table-cell	此元素会作为一个表格单元格显示（类似 <td> 和 <th>）
table-caption	此元素会作为一个表格标题显示（类似 <caption>）
inherit	规定应该从父元素继承 display 属性的值。

TIY 实例

如何把元素显示为内联元素

本例演示如何把元素显示为内联元素。

```
<html>
<head>
<style type="text/css">
p {display: inline}
div {display: none}
</style>
</head>

<body>
<p>本例中的样式表把段落元素设置为内联元素。</p>

<p>而 div 元素不会显示出来！</p>

<div>div 元素的内容不会显示出来！</div>
</body>
</html>
```

如何把元素显示为块级元素

本例演示如何把元素显示为块级元素。

```
<html>
<head>
<style type="text/css">
span
{
display: block
}
</style>
</head>
<body>

<span>本例中的样式表把 span 元素设置为块级元素。</span>
<span>两个 span 元素之间产生了一个换行行为。</span>

</body>
</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：display 属性

CSS float 属性

实例

把图像向右浮动：

```
img
  {
  float:right;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 float 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

float 属性定义元素在哪个方向浮动。以往这个属性总应用于图像，使文本围绕在图像周围，不过在 CSS 中，任何元素都可以浮动。浮动元素会生成一个块级框，而不论它本身是何种元素。

如果浮动非替换元素，则要指定一个明确的宽度；否则，它们会尽可能地窄。

注释：假如在一行之上只有极少的空间可供浮动元素，那么这个元素会跳至下一行，这个过程会持续到某一行拥有足够的空间为止。

默认值：	**none**
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.cssFloat="left"

可能的值

值	描述
left	元素向左浮动。
right	元素向右浮动。
none	默认值。元素不浮动，并会显示在其在文本中出现的位置。
inherit	规定应该从父元素继承 float 属性的值。

TIY 实例

float 属性的简单应用

使图像浮动于一个段落的右侧。

```
<html>
<head>
<style type="text/css">
img
{
float:right
}
</style>
</head>

<body>
<p>在下面的段落中，我们添加了一个样式为 <b>float:right</b> 的图像。结果是这个图像会浮动到段落的右侧。<
<p>
<img src="/i/eg_cute.gif" />
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>
</body>

</html>
```

将带有边框和边界的图像浮动于段落的右侧

使图像浮动于段落的右侧。向图像添加边框和边界。

```
<html>
<head>
<style type="text/css">
img
{
float:right;
border:1px dotted black;
margin:0px 0px 15px 20px;
}
</style>
</head>

<body>
<p>在下面的段落中，图像会浮动到右侧，并且添加了点状的边框。我们还为图像添加了边距，这样就可以把文本推离图像
<p>
<img src="/i/eg_cute.gif" />
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>
</body>

</html>
```

带标题的图像浮动于右侧

使带有标题的图像浮动于右侧

```
<html>
<head>
<style type="text/css">
div
{
float:right;
width:120px;
margin:0 0 15px 20px;
padding:15px;
border:1px solid black;
text-align:center;
}
</style>
</head>

<body>
<div>
<img src="/i/eg_cute.gif" /><br />
CSS is fun!
</div>
<p>
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>

<p>
在上面的段落中，div 元素的宽度是 120 像素，它其中包含图像。div 元素浮动到右侧。我们向 div 元素添加了外
</p>
</body>

</html>
```

使段落的首字母浮动于左侧

使段落的首字母浮动于左侧，并向这个字母添加样式。

```
<html>
<head>
<style type="text/css">
span
{
float:left;
width:0.7em;
font-size:400%;
font-family:algerian,courier;
line-height:80%;
}
</style>
</head>

<body>
<p>
<span>T</span>his is some text.
This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>

<p>
在上面的段落中，文本的第一个字母包含在一个 span 元素中。这个 span 元素的宽度是当前字体尺寸的 0.7 倍。sp
</p>

</body>
</html>
```

创建水平菜单

使用具有一栏超链接的浮动来创建水平菜单。

```
<html>
<head>
<style type="text/css">
ul
{
float:left;
width:100%;
padding:0;
margin:0;
list-style-type:none;
}
a
{
float:left;
width:7em;
text-decoration:none;
color:white;
background-color:purple;
padding:0.2em 0.6em;
border-right:1px solid white;
}
a:hover {background-color:#ff3300}
li {display:inline}
</style>
</head>

<body>
<ul>
<li><a href="#">Link one</a></li>
<li><a href="#">Link two</a></li>
<li><a href="#">Link three</a></li>
<li><a href="#">Link four</a></li>
</ul>

<p>
在上面的例子中，我们把 ul 元素和 a 元素浮向左浮动。li 元素显示为行内元素（元素前后没有换行）。这样就可以
</p>

</body>
</html>
```



创建无表格的首页

使用浮动来创建拥有页眉、页脚、左侧目录和主体内容的首页。

```
<html>
<head>
<style type="text/css">
div.container
{
width:100%;
margin:0px;
border:1px solid gray;
line-height:150%;
}
div.header,div.footer
{
padding:0.5em;
color:white;
background-color:gray;
clear:left;
}
h1.header
{
padding:0;
margin:0;
}
div.left
{
float:left;
width:160px;
margin:0;
padding:1em;
}
div.content
{
margin-left:190px;
border-left:1px solid gray;
padding:1em;
}
</style>
</head>
<body>

<div class="container">

<div class="header"><h1 class="header">W3School.com.cn</h1></div>

<div class="left"><p>"Never increase, beyond what is necessary, the number of entities re

<div class="content">
<h2>Free Web Building Tutorials</h2>
<p>At W3School.com.cn you will find all the Web-building tutorials you need,
from basic HTML and XHTML to advanced XML, XSL, Multimedia and WAP.</p>
<p>W3School.com.cn - The Largest Web Developers Site On The Net!</p></div>

<div class="footer">Copyright 2008 by YingKe Investment.</div>
</div>

</body>
</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：cssFloat 属性

CSS left 属性

实例

把图像的左边缘设置在其包含元素左边缘向右 100 像素的位置：

```
img
  {
  position:absolute;
  left:100px;
  }
```

浏览器支持

所有主流浏览器都支持 left 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

left 属性规定元素的左边缘。该属性定义了定位元素左外边距边界与其包含块左边界之间的偏移。

注释：如果 "position" 属性的值为 "static"，那么设置 "left" 属性不会产生任何效果。

说明

对于 static 元素，为 auto；对于长度值，则为相应的绝对长度；对于百分比数值，为指定值；否则为 auto。

对于相对定义元素，left 的计算值始终等于 right。

默认值：	**auto**
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.left="100px"

可能的值

值	描述
auto	默认值。通过浏览器计算左边缘的位置。
%	设置以包含元素的百分比计的左边位置。可使用负值。
length	使用 px、cm 等单位设置元素的左边位置。可使用负值。
inherit	规定应该从父元素继承 left 属性的值。

TIY 实例

使用固定值设置图像的左边缘

本例演示如何使用固定值设置图像的左边缘。

```
<html>
<head>
<style type="text/css">
img
{
position:absolute;
left:100px
}
</style>
</head>
<body>

<h1>这是标题</h1>
<img class="normal" src="/i/eg_smile.gif" />
<p>一些文本。一些文本。一些文本。一些文本。一些文本。一些文本。</p>

</body>
</html>
```

使用百分比设置图像的左边缘

本例演示如何使用百分比值设置图像的左边缘。

```
<html>
<head>
<style type="text/css">
img
{
position:absolute;
left:20%
}
</style>
</head>
<body>

<h1>这是标题</h1>
<img class="normal" src="/i/eg_smile.gif" />
<p>一些文本。一些文本。一些文本。一些文本。一些文本。一些文本。</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：left 属性

CSS overflow 属性

实例

设置 overflow 属性：

```
div
  {
  width:150px;
  height:150px;
  overflow:scroll;
  }
```

浏览器支持

所有主流浏览器都支持 overflow 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

overflow 属性规定当内容溢出元素框时发生的事情。

说明

这个属性定义溢出元素内容区的内容会如何处理。如果值为 scroll，不论是否需要，用户代理都会提供一种滚动机制。因此，有可能即使元素框中可以放下所有内容也会出现滚动条。

默认值：	**visible**
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.overflow="scroll"

可能的值

值	描述
visible	默认值。内容不会被修剪，会呈现在元素框之外。
hidden	内容会被修剪，并且其余内容是不可见的。
scroll	内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。
auto	如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。
inherit	规定应该从父元素继承 overflow 属性的值。

TIY 实例

如何使用滚动条来显示元素内溢出的内容

本例演示当元素内容太大而超出规定区域时，如何设置溢出属性来规定相应的动作。

```
<html>
<head>
<style type="text/css">
div
{
background-color:#00FFFF;
width:150px;
height:150px;
overflow: scroll
}
</style>
</head>

<body>
<p>如果元素中的内容超出了给定的宽度和高度属性，overflow 属性可以确定是否显示滚动条等行为。</p>

<div>
这个属性定义溢出元素内容区的内容会如何处理。如果值为 scroll，不论是否需要，用户代理都会提供一种滚动机制。
</div>
</body>

</html>
```

如何隐藏溢出元素中溢出的内容

本例演示在元素中的内容太大以至于无法适应指定的区域时，如何设置 overflow 属性来隐藏其内容。

```
<html>
<head>
<style type="text/css">
div
{
background-color:#00FFFF;
width:150px;
height:150px;
overflow: hidden
}
</style>
</head>

<body>
<p>如果元素中的内容超出了给定的宽度和高度属性，overflow 属性可以确定是否显示滚动条等行为。</p>

<div>
这个属性定义溢出元素内容区的内容会如何处理。如果值为 scroll，不论是否需要，用户代理都会提供一种滚动机制。
</div>
</body>

</html>
```

如何设置浏览器来自动地处理溢出

本例演示如何设置浏览器来自动地处理溢出。

```
<html>
<head>
<style type="text/css">
div
{
background-color:#00FFFF;
width:150px;
height:150px;
overflow: auto
}
</style>
</head>

<body>
<p>如果元素中的内容超出了给定的宽度和高度属性，overflow 属性可以确定是否显示滚动条等行为。</p>

<div>
这个属性定义溢出元素内容区的内容会如何处理。如果值为 scroll，不论是否需要，用户代理都会提供一种滚动机制。
</div>
</body>

</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：overflow 属性

CSS position 属性

实例

定位 h2 元素：

```
h2
  {
  position:absolute;
  left:100px;
  top:150px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 position 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

position 属性规定元素的定位类型。

说明

这个属性定义建立元素布局所用的定位机制。任何元素都可以定位，不过绝对或固定元素会生成一个块级框，而不论该元素本身是什么类型。相对定位元素会相对于它在正常流中的默认位置偏移。

默认值：	static
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.position="absolute"

可能的值

值	描述
absolute	

生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。

元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。

| | fixed |

生成绝对定位的元素，相对于浏览器窗口进行定位。

元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。

| | relative |

生成相对定位的元素，相对于其正常位置进行定位。

因此，"left:20" 会向元素的 LEFT 位置添加 20 像素。

| | static | 默认值。没有定位，元素出现在正常的流中（忽略 top, bottom, left, right 或者 z-index 声明）。 | | inherit | 规定应该从父元素继承 position 属性的值。 |

TIY 实例

定位：相对定位

本例演示如何相对于一个元素的正常位置来对其定位。

```
<html>
<head>
<style type="text/css">
h2.pos_left
{
position:relative;
left:-20px
}
h2.pos_right
{
position:relative;
left:20px
}
</style>
</head>

<body>
<h2>这是位于正常位置的标题</h2>
<h2 class="pos_left">这个标题相对于其正常位置向左移动</h2>
<h2 class="pos_right">这个标题相对于其正常位置向右移动</h2>
<p>相对定位会按照元素的原始位置对该元素进行移动。</p>
<p>样式 "left:-20px" 从元素的原始左侧位置减去 20 像素。</p>
<p>样式 "left:20px" 向元素的原始左侧位置增加 20 像素。</p>
</body>

</html>
```

定位：绝对定位

本例演示如何使用绝对值来对元素进行定位。

```
<html>
<head>
<style type="text/css">
h2.pos_abs
{
position:absolute;
left:100px;
top:150px
}
</style>
</head>

<body>
<h2 class="pos_abs">这是带有绝对定位的标题</h2>
<p>通过绝对定位，元素可以放置到页面上的任何位置。下面的标题距离页面左侧 100px，距离页面顶部 150px。</p>
</body>

</html>
```

定位：固定定位

本例演示如何相对于浏览器窗口来对元素进行定位。

```
<html>
<head>
<style type="text/css">
p.one
{
position:fixed;
left:5px;
top:5px;
}
p.two
{
position:fixed;
top:30px;
right:5px;
}
</style>
</head>
<body>

<p class="one">一些文本。</p>
<p class="two">更多的文本。</p>

</body>
</html>
```

设置元素的形状

本例演示如何设置元素的形状。此元素被剪裁到这个形状内，并显示出来。

```
<html>
<head>
<style type="text/css">
img
{
position:absolute;
clip:rect(0px 50px 200px 0px)
}
</style>
</head>

<body>
<p>clip 属性剪切了一幅图像：</p>
<p><img border="0" src="/i/eg_bookasp.gif" width="120" height="151"></p>
</body>

</html>
```

Z-index

Z-index可被用于将在一个元素放置于另一元素之后。

```
<html>
<head>
<style type="text/css">
img.x
{
position:absolute;
left:0px;
top:0px;
z-index:-1
}
</style>
</head>

<body>
<h1>这是一个标题</h1>
<img class="x" src="/i/eg_mouse.jpg" />
<p>默认的 z-index 是 0。Z-index -1 拥有更低的优先级。</p>
</body>

</html>
```

Z-index

上面的例子中的元素已经更改了Z-index。

```
<html>
<head>
<style type="text/css">
img.x
{
position:absolute;
left:0px;
top:0px;
z-index:1
}
</style>
</head>

<body>
<h1>这是一个标题</h1>
<img class="x" src="/i/eg_mouse.jpg" />
<p>默认的 z-index 是 0。Z-index 1 拥有更高的优先级。</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：position 属性

CSS right 属性

实例

把图像的右边缘设置在其包含元素右边缘向左 5 像素的位置：

```
img
  {
  position:absolute;
  right:5px;
  }
```

浏览器支持

所有主流浏览器都支持 right 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

right 属性规定元素的右边缘。该属性定义了定位元素右外边距边界与其包含块右边界之间的偏移。

注释：如果 "position" 属性的值为 "static"，那么设置 "right" 属性不会产生任何效果。

说明

对于 static 元素，为 auto；对于长度值，则为相应的绝对长度；对于百分比数值，为指定值；否则为 auto。

对于相对定义元素，left 的计算值始终等于 right。

默认值：	**auto**
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.right="50px"

可能的值

值	描述
auto	默认值。通过浏览器计算右边缘的位置。
%	设置以包含元素的百分比计的右边位置。可使用负值。
length	使用 px、cm 等单位设置元素的右边位置。可使用负值。
inherit	规定应该从父元素继承 right 属性的值。

TIY 实例

使用固定值设置图像的右边缘

本例演示如何使用固定值设置图像的右边缘。

```
<html>
<head>
<style type="text/css">
img
{
position:absolute;
right:0px
}
</style>
</head>
<body>

<h1>这是标题</h1>
<img class="normal" src="/i/eg_smile.gif" />
<p>一些文本。一些文本。一些文本。一些文本。一些文本。一些文本。</p>

</body>
</html>
```

使用百分比设置图像的右边缘

本例演示如何使用百分比值设置图像的右边缘。

```
<html>
<head>
<style type="text/css">
img
{
position:absolute;
right:5%
}
</style>
</head>
<body>

<h1>这是标题</h1>
<img class="normal" src="/i/eg_smile.gif" />
<p>一些文本。一些文本。一些文本。一些文本。一些文本。一些文本。</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：right 属性

CSS top 属性

实例

把图像的上边缘设置在其包含元素上边缘之下 5 像素高的位置：

```
img
  {
  position:absolute;
  top:5px;
  }
```

浏览器支持

所有主流浏览器都支持 top 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

top 属性规定元素的顶部边缘。该属性定义了一个定位元素的上外边距边界与其包含块上边界之间的偏移。

注释：如果 "position" 属性的值为 "static"，那么设置 "top" 属性不会产生任何效果。

说明

对于 static 元素，为 auto；对于长度值，则为相应的绝对长度；对于百分比数值，为指定值；否则为 auto。

对于相对定义元素，如果 top 和 bottom 都是 auto，其计算值则都是 0；如果其中之一为 auto，则取另一个值的相反数；如果二者都不是 auto，bottom 将取 top 值的相反数。

默认值：	auto
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.top="50px"

可能的值

值	描述
auto	默认值。通过浏览器计算上边缘的位置。
%	设置以包含元素的百分比计的上边位置。可使用负值。
length	使用 px、cm 等单位设置元素的上边位置。可使用负值。
inherit	规定应该从父元素继承 top 属性的值。

TIY 实例

使用固定值设置图像的上边缘

本例演示如何使用固定值设置图像的上边缘。

```
<html>
<head>
<style type="text/css">
img
{
position:absolute;
top:0px
}
</style>
</head>
<body>

<h1>This is a Heading</h1>
<img class="normal" src="/i/eg_smile.gif" />
<p>一些文本。一些文本。一些文本。一些文本。一些文本。一些文本。</p>

</body>
</html>
```

使用百分比设置图像的上边缘

本例演示如何使用百分比值设置图像的上边缘。

```
<html>
<head>
<style type="text/css">
img
{
position:absolute;
top:5%
}
</style>
</head>
<body>

<h1>这是标题</h1>
<img class="normal" src="/i/eg_smile.gif" />
<p>一些文本。一些文本。一些文本。一些文本。一些文本。一些文本。</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：top 属性

CSS vertical-align 属性

实例

垂直对齐一幅图像：

```
img
  {
  vertical-align:text-top;
  }
```

浏览器支持

所有浏览器都支持 vertical-align 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

vertical-align 属性设置元素的垂直对齐方式。

说明

该属性定义行内元素的基线相对于该元素所在行的基线的垂直对齐。允许指定负长度值和百分比值。这会使元素降低而不是升高。在表单元格中，这个属性会设置单元格框中的单元格内容的对齐方式。

默认值：	**baseline**
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.verticalAlign="bottom"

可能的值

值	描述
baseline	默认。元素放置在父元素的基线上。
sub	垂直对齐文本的下标。
super	垂直对齐文本的上标
top	把元素的顶端与行中最高元素的顶端对齐
text-top	把元素的顶端与父元素字体的顶端对齐
middle	把此元素放置在父元素的中部。
bottom	把元素的顶端与行中最低的元素的顶端对齐。
text-bottom	把元素的底端与父元素字体的底端对齐。
length	
%	使用 "line-height" 属性的百分比值来排列此元素。允许使用负值。
inherit	规定应该从父元素继承 vertical-align 属性的值。

TIY 实例

垂直对齐图象

本例演示如何在文本中垂直排列图象。

```
<html>

<head>
<style type="text/css">
img.top {vertical-align:text-top}
img.bottom {vertical-align:text-bottom}
</style>
</head>

<body>

<p>
这是一幅<img class="top" border="0" src="/i/eg_cute.gif" />位于段落中的图像。
</p>

<p>
这是一幅<img class="bottom" border="0" src="/i/eg_cute.gif" />位于段落中的图像。
</p>

</body>

</html>
```

相关页面

CSS 教程：CSS 文本

HTML DOM 参考手册：verticalAlign 属性

CSS visibility 属性

实例

使 h2 元素不可见：

```
h2
  {
  visibility:hidden;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 visibility 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持 "inherit" 和 "collapse" 属性值。

定义和用法

visibility 属性规定元素是否可见。

提示：即使不可见的元素也会占据页面上的空间。请使用 "display" 属性来创建不占据页面空间的不可见元素。

说明

这个属性指定是否显示一个元素生成的元素框。这意味着元素仍占据其本来的空间，不过可以完全不可见。值 collapse 在表中用于从表布局中删除列或行。

默认值：	visible
继承性：	yes
版本：	CSS2
JavaScript 语法：	*object*.style.visibility="hidden"

可能的值

值	描述
visible	默认值。元素是可见的。
hidden	元素是不可见的。
collapse	当在表格元素中使用时，此值可删除一行或一列，但是它不会影响表格的布局。被行或列占据的空间会留给其他内容使用。如果此值被用在其他的元素上，会呈现为 "hidden"。
inherit	规定应该从父元素继承 visibility 属性的值。

TIY 实例

如何使元素不可见

本例演示如何使元素不可见。你希望元素被显示出来，还是不呢？

```
<html>
<head>
<style type="text/css">
h1.visible {visibility:visible}
h1.invisible {visibility:hidden}
</style>
</head>

<body>
<h1 class="visible">这是可见的标题</h1>
<h1 class="invisible">这是不可见的标题</h1>
</body>

</html>
```

把表格元素设置为 collapse

本例演示如何使表格元素叠加？

```
<html>
<head>
<style type="text/css">
tr.coll
 {
 visibility:collapse
 }
</style>
</head>
<body>

<table border="1">
<tr>
<td>Adams</td>
<td>John</td>
</tr>
<tr class="coll">
<td>Bush</td>
<td>George</td>
</tr>
</table>

</body>
</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：visibility 属性

CSS z-index 属性

实例

设置图像的 z-index：

```
img
  {
  position:absolute;
  left:0px;
  top:0px;
  z-index:-1;
  }
```

浏览器支持

所有主流浏览器都支持 z-index 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

z-index 属性设置元素的堆叠顺序。拥有更高堆叠顺序的元素总是会处于堆叠顺序较低的元素的前面。

注释：元素可拥有负的 z-index 属性值。

注释：Z-index 仅能在定位元素上奏效（例如 position:absolute;）！

说明

该属性设置一个定位元素沿 z 轴的位置，z 轴定义为垂直延伸到显示区的轴。如果为正数，则离用户更近，为负数则表示离用户更远。

默认值：	auto
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.zIndex="1"

可能的值

值	描述
auto	默认。堆叠顺序与父元素相等。
number	设置元素的堆叠顺序。
inherit	规定应该从父元素继承 z-index 属性的值。

TIY 实例

Z-index

Z-index 可用于将在一个元素放置于另一元素之后。

```
<html>
<head>
<style type="text/css">
img.x
{
position:absolute;
left:0px;
top:0px;
z-index:-1
}
</style>
</head>

<body>
<h1>这是一个标题</h1>
<img class="x" src="/i/eg_mouse.jpg" />
<p>默认的 z-index 是 0。Z-index -1 拥有更低的优先级。</p>
</body>

</html>
```

Z-index

上例中的元素已经更改了 Z-index。

```
<html>
<head>
<style type="text/css">
img.x
{
position:absolute;
left:0px;
top:0px;
z-index:1
}
</style>
</head>

<body>
<h1>这是一个标题</h1>
<img class="x" src="/i/eg_mouse.jpg" />
<p>默认的 z-index 是 0。Z-index 1 拥有更高的优先级。</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 定位

HTML DOM 参考手册：zIndex 属性

CSS 打印属性（Print）

属性	描述	CSS
orphans	设置当元素内部发生分页时必须在页面底部保留的最少行数。	2
page-break-after	设置元素后的分页行为。	2
page-break-before	设置元素前的分页行为。	2
page-break-inside	设置元素内部的分页行为。	2
widows	设置当元素内部发生分页时必须在页面顶部保留的最少行数。	2

CSS page-break-after 属性

实例

设置在表格元素之后始终进行分页的分页行为：

```
<html>
<head>
<style>
@media print
{
table {page-break-after:always;}
}
</style>
</head>

<body>
....
</body>
</html>
```

浏览器支持

所有浏览器都支持 page-break-after 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "left"、"right" 以及 "inherit"。

注释：Firefox、Chrome 以及 Safari 不支持属性值 "avoid"、"left" 以及 "right"。

定义和用法

page-break-after 属性设置元素后的 page-breaking 行为。

尽管可以用 always 强制放上分页符，但是无法保证避免分页符的插入，创作人员最多只能要求用户代理尽可能避免插入分页。

应用于：position 值为 relative 或 static 的非浮动块级元素。

注释：请尽可能少地使用分页属性，并且避免在表格、浮动元素、带有边框的块元素中使用分页属性。

默认值：	**auto**
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.pageBreakAfter="always"

可能的值

值	描述
auto	默认。如果必要则在元素后插入分页符。
always	在元素后插入分页符。
avoid	避免在元素后插入分页符。
left	在元素之后足够的分页符，一直到一张空白的左页为止。
right	在元素之后足够的分页符，一直到一张空白的右页为止。
inherit	规定应该从父元素继承 page-break-after 属性的设置。

相关页面

HTML DOM 参考手册：pageBreakAfter 属性

CSS page-break-before 属性

实例

设置在表格元素之后始终进行分页的分页行为：

```
<html>
<head>
<style>
@media print
{
table {page-break-before:always;}
}
</style>
</head>

<body>
....
</body>
</html>
```

浏览器支持

所有浏览器都支持 page-break-before 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "left"、"right" 以及 "inherit"。

注释：Firefox、Chrome 以及 Safari 不支持属性值 "avoid"、"left" 以及 "right"。

定义和用法

page-break-before 属性设置元素前的 page-breaking 行为。

尽管可以用 always 强制放上分页符，但是无法保证避免分页符的插入，创作人员最多只能要求用户代理尽可能避免插入分页。

应用于：position 值为 relative 或 static 的非浮动块级元素。

注释：请尽可能少地使用分页属性，并且避免在表格、浮动元素、带有边框的块元素中使用分页属性。

默认值：	**auto**
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.pageBreakBefore="always"

可能的值

值	描述
auto	默认值。如果必要则在元素前插入分页符。
always	在元素前插入分页符。
avoid	避免在元素前插入分页符。
left	在元素之前足够的分页符，一直到一张空白的左页为止。
right	在元素之前足够的分页符，一直到一张空白的右页为止。
inherit	规定应该从父元素继承 page-break-before 属性的设置。

相关页面

HTML DOM 参考手册：pageBreakBefore 属性

CSS page-break-inside 属性

实例

设置在表格元素内部避免进行分页的分页行为：

```
<html>
<head>
<style>
@media print
{
table {page-break-inside:avoid;}
}
</style>
</head>

<body>
....
</body>
</html>
```

浏览器支持

只有 Opera 浏览器支持 page-break-inside 属性。

定义和用法

page-break-inside 属性设置元素内部的 page-breaking 行为。

尽管可以用 always 强制放上分页符，但是无法保证避免分页符的插入，创作人员最多只能要求用户代理尽可能避免插入分页。

应用于：position 值为 relative 或 static 的非浮动块级元素。

注释：请尽可能少地使用分页属性，并且避免在表格、浮动元素、带有边框的块元素中使用分页属性。

默认值：	**auto**
继承性：	no
版本：	CSS2
JavaScript 语法：	*object*.style.pageBreakInside="avoid"

可能的值

值	描述
auto	默认。如果必要则在元素内部插入分页符。
avoid	避免在元素内部插入分页符。
inherit	规定应该从父元素继承 page-break-inside 属性的设置。

相关页面

HTML DOM 参考手册：pageBreakInside 属性

CSS 表格属性（Table）

属性	描述	CSS
border-collapse	规定是否合并表格边框。	2
border-spacing	规定相邻单元格边框之间的距离。	2
caption-side	规定表格标题的位置。	2
empty-cells	规定是否显示表格中的空单元格上的边框和背景。	2
table-layout	设置用于表格的布局算法。	2

CSS border-collapse 属性

实例

为表格设置合并边框模型：

```
table
  {
  border-collapse:collapse;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 border-collapse 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

注释：如果没有规定 !DOCTYPE，则 border-collapse 可能产生意想不到的结果。

定义和用法

border-collapse 属性设置表格的边框是否被合并为一个单一的边框，还是象在标准的 HTML 中那样分开显示。

默认值：	**separate**
继承性：	yes
---	---
版本：	CSS2
---	---
JavaScript 语法：	*object*.style.borderCollapse="collapse"
---	---

可能的值

值	描述
separate	默认值。边框会被分开。不会忽略 border-spacing 和 empty-cells 属性。
collapse	如果可能，边框会合并为一个单一的边框。会忽略 border-spacing 和 empty-cells 属性。
inherit	规定应该从父元素继承 border-collapse 属性的值。

TIY 实例

合并表格边框

本例演示是否把表格边框显示为一条单独的边框，还是像标准的 HTML 中那样分开显示。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtm
<html>
<head>
<style type="text/css">
table
  {
  border-collapse:collapse;
  }

table, td, th
  {
  border:1px solid black;
  }
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
</tr>
<tr>
<td>Bill</td>
<td>Gates</td>
</tr>
<tr>
<td>Steven</td>
<td>Jobs</td>
</tr>
</table>
<p><b>注释：</b>如果没有规定 !DOCTYPE，border-collapse 属性可能会引起意想不到的错误。</p>
</body>
</html>
```

相关页面

CSS 教程：CSS 表格

HTML DOM 参考手册：borderCollapse 属性

CSS border-spacing 属性

实例

为表格设置 border-spacing：

```
table
  {
  border-collapse:separate;
  border-spacing:10px 50px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 border-spacing 属性。

注释：如果已规定 !DOCTYPE，那么 Internet Explorer 8 （以及更高版本）支持 border-spacing 属性。

定义和用法

border-spacing 属性设置相邻单元格的边框间的距离（仅用于“边框分离”模式）。

注释：某些版本的IE浏览器不支持此属性。

说明

该属性指定分隔边框模型中单元格边界之间的距离。在指定的两个长度值中，第一个是水平间隔，第二个是垂直间隔。除非 border-collapse 被设置为 separate，否则将忽略这个属性。尽管这个属性只应用于表，不过它可以由表中的所有元素继承。

默认值：	***not specified***
继承性：	yes
版本：	CSS2
JavaScript 语法：	*object*.style.borderSpacing="15px"

可能的值

值	描述
length length	

规定相邻单元的边框之间的距离。使用 px、cm 等单位。不允许使用负值。

如果定义一个 *length* 参数，那么定义的是水平和垂直间距。

如果定义两个 *length* 参数，那么第一个设置水平间距，而第二个设置垂直间距。

| | inherit | 规定应该从父元素继承 border-spacing 属性的值。 |

TIY 实例

设置表格边框之间的空白

本例演示如何设置单元格边框之间的距离。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtm

<html>
<head>
<style type="text/css">
table.one
{
border-collapse: separate;
border-spacing: 10px
}
table.two
{
border-collapse: separate;
border-spacing: 10px 50px
}
</style>
</head>
<body>

<table class="one" border="1">
<tr>
<td>Adams</td>
<td>John</td>
</tr>
<tr>
<td>Bush</td>
<td>George</td>
</tr>
</table>

<br />

<table class="two" border="1">
<tr>
<td>Carter</td>
<td>Thomas</td>
</tr>
<tr>
<td>Gates</td>
<td>Bill</td>
</tr>
</table>

<p><b>注释：</b>如果已规定 !DOCTYPE，那么 Internet Explorer 8 （以及更高版本）支持 border-spacin

</body>
</html>
```

相关页面

CSS 教程：CSS 表格

HTML DOM 参考手册：borderSpacing 属性

CSS caption-side 属性

实例

规定表格标题的放置方式：

```
caption
  {
  caption-side:bottom;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 caption-side 属性。

注释：如果已规定 !DOCTYPE，那么 Internet Explorer 8 （以及更高版本）支持 caption-side 属性。

定义和用法

caption-side 属性设置表格标题的位置。

说明

该属性指定了表标题相对于表框的放置位置。表标题显示为好像它是表之前（或之后）的一个块级元素。

默认值：	top
继承性：	yes
版本：	CSS2
JavaScript 语法：	*object*.style.captionSide="bottom"

可能的值

值	描述
top	默认值。把表格标题定位在表格之上。
bottom	把表格标题定位在表格之下。
inherit	规定应该从父元素继承 caption-side 属性的值。

TIY 实例

设置表格标题的位置

本例演示如何定位表格的标题。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtm

<html>
<head>
<style type="text/css">
caption
{
caption-side:bottom
}
</style>
</head>
<body>

<table border="1">
<caption>This is a caption</caption>
<tr>
<td>Adams</td>
<td>John</td>
</tr>
<tr>
<td>Bush</td>
<td>George</td>
</tr>
</table>

<p><b>注释：</b>如果已规定 !DOCTYPE，那么 Internet Explorer 8 （以及更高版本）支持 caption-side

</body>
</html>
```

相关页面

CSS 教程：CSS 表格

HTML DOM 参考手册：captionSide 属性

CSS empty-cells 属性

实例

隐藏表格中空单元格上的边框和背景：

```
table
  {
  border-collapse:separate;
  empty-cells:hide;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 empty-cells 属性。

注释：如果已规定 !DOCTYPE，那么 Internet Explorer 8 （以及更高版本）支持 empty-cells 属性。

定义和用法

empty-cells 属性设置是否显示表格中的空单元格（仅用于“分离边框”模式）。

注释：某些版本的 IE 浏览器不支持此属性。

说明

该属性定义了不包含任何内容的表单元格如何表示。如果显示，就会绘制出单元格的边框和背景。除非 border-collapse 设置为 separate，否则将忽略这个属性。

默认值：	show
继承性：	yes
版本：	CSS2
JavaScript 语法：	*object*.style.emptyCells="hide"

可能的值

值	描述
hide	不在空单元格周围绘制边框。
show	在空单元格周围绘制边框。默认。
inherit	规定应该从父元素继承 empty-cells 属性的值。

相关页面

CSS 教程：CSS 表格

HTML DOM 参考手册：emptyCells 属性

CSS table-layout 属性

实例

设置表格布局算法：

```
table
  {
  table-layout:fixed;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 table-layout 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

tableLayout 属性用来显示表格单元格、行、列的算法规则。

固定表格布局：

固定表格布局与自动表格布局相比，允许浏览器更快地对表格进行布局。

在固定表格布局中，水平布局仅取决于表格宽度、列宽度、表格边框宽度、单元格间距，而与单元格的内容无关。

通过使用固定表格布局，用户代理在接收到第一行后就可以显示表格。

自动表格布局：

在自动表格布局中，列的宽度是由列单元格中没有折行的最宽的内容设定的。

此算法有时会较慢，这是由于它需要在确定最终的布局之前访问表格中所有的内容。

说明

该属性指定了完成表布局时所用的布局算法。固定布局算法比较快，但是不太灵活，而自动算法比较慢，不过更能反映传统的 HTML 表。

默认值：	**auto**
继承性：	yes
版本：	CSS2
JavaScript 语法：	*object*.style.tableLayout="fixed"

可能的值

值	描述
automatic	默认。列宽度由单元格内容设定。
fixed	列宽由表格宽度和列宽度设定。
inherit	规定应该从父元素继承 table-layout 属性的值。

TIY 实例

设置表格的布局

本例演示如何设置表格的布局。

```
<html>
<head>
<style type="text/css">
table.one
{
table-layout: automatic
}
table.two
{
table-layout: fixed
}
</style>
</head>
<body>

<table class="one" border="1" width="100%">
<tr>
<td width="20%">1000000000000000000000000000</td>
<td width="40%">10000000</td>
<td width="40%">100</td>
</tr>
</table>

<br />

<table class="two" border="1" width="100%">
<tr>
<td width="20%">1000000000000000000000000000</td>
<td width="40%">10000000</td>
<td width="40%">100</td>
</tr>
</table>

</body>
</html>
```

相关页面

CSS 教程：CSS 表格

HTML DOM 参考手册：tableLayout 属性

CSS 文本属性（Text）

属性	描述	CSS
color	设置文本的颜色。	1
direction	规定文本的方向 / 书写方向。	2
letter-spacing	设置字符间距。	1
line-height	设置行高。	1
text-align	规定文本的水平对齐方式。	1
text-decoration	规定添加到文本的装饰效果。	1
text-indent	规定文本块首行的缩进。	1
text-shadow	规定添加到文本的阴影效果。	2
text-transform	控制文本的大小写。	1
unicode-bidi	设置文本方向。	2
white-space	规定如何处理元素中的空白。	1
word-spacing	设置单词间距。	1
hanging-punctuation	规定标点字符是否位于线框之外。	3
punctuation-trim	规定是否对标点字符进行修剪。	3
text-align-last	设置如何对齐最后一行或紧挨着强制换行符之前的行。	3
text-emphasis	向元素的文本应用重点标记以及重点标记的前景色。	3
text-justify	规定当 text-align 设置为 "justify" 时所使用的对齐方法。	3
text-outline	规定文本的轮廓。	3
text-overflow	规定当文本溢出包含元素时发生的事情。	3
text-shadow	向文本添加阴影。	3
text-wrap	规定文本的换行规则。	3
word-break	规定非中日韩文本的换行规则。	3
word-wrap	允许对长的不可分割的单词进行分割并换行到下一行。	3

CSS color 属性

实例

为不同元素设置文本颜色：

```
body
  {
  color:red;
  }
h1
  {
  color:#00ff00;
  }
p
  {
  color:rgb(0,0,255);
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 color 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

color 属性规定文本的颜色。

这个属性设置了一个元素的前景色（在 HTML 表现中，就是元素文本的颜色）；光栅图像不受 color 影响。这个颜色还会应用到元素的所有边框，除非被 border-color 或另外某个边框颜色属性覆盖。

要设置一个元素的前景色，最容易的方法是使用 color 属性。

默认值：	***not specified***
继承性：	yes
版本：	CSS1
JavaScript 语法：	*object*.style.color="#FF0000"

提示和注释

提示：请使用合理的背景颜色和文本颜色搭配，这样可以提高文本的可读性。

可能的值

值	描述
color_name	规定颜色值为颜色名称的颜色（比如 red）。
hex_number	规定颜色值为十六进制值的颜色（比如 #ff0000）。
rgb_number	规定颜色值为 rgb 代码的颜色（比如 rgb(255,0,0)）。
inherit	规定应该从父元素继承颜色。

TIY 实例

设置文本颜色

本例演示如何设置文本的颜色。

```
<html>
<head>
<style type="text/css">
body {color:red}
h1 {color:#00ff00}
p.ex {color:rgb(0,0,255)}
</style>
</head>

<body>
<h1>这是 heading 1</h1>
<p>这是一段普通的段落。请注意，该段落的文本是红色的。在 body 选择器中定义了本页面中的默认文本颜色。</p>
<p class="ex">该段落定义了 class="ex"。该段落中的文本是蓝色的。</p>
</body>
</html>
```

相关页面

CSS 教程：CSS 文本

HTML DOM 参考手册：color 属性

CSS direction 属性

实例

把文本方向设置为“从右向左”：

```
div
  {
  direction: rtl
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 direction 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

direction 属性规定文本的方向 / 书写方向。

该属性指定了块的基本书写方向，以及针对 Unicode 双向算法的嵌入和覆盖方向。不支持双向文本的用户代理可以忽略这个属性。

默认值：	ltr
继承性：	yes
版本：	CSS2
JavaScript 语法：	*object*.style.direction="rtl"

可能的值

值	描述
ltr	默认。文本方向从左到右。
rtl	文本方向从右到左。
inherit	规定应该从父元素继承 direction 属性的值。

TIY 实例

设置文本的方向

本例演示如何设置文本的方向。

```
<html>
<head>
<style type="text/css">
div.one
{
direction: rtl
}
div.two
{
direction: ltr
}
</style>
</head>
<body>

<div class="one">Some text. Right-to-left direction.</div>
<div class="two">Some text. Left-to-right direction.</div>

</body>
</html>
```

相关页面

CSS 教程：CSS 文本

HTML DOM 参考手册：direction 属性

CSS letter-spacing 属性

实例

设置 h1 和 h2 元素的字母间距：

```
h1 {letter-spacing:2px}
h2 {letter-spacing:-3px}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 letter-spacing 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

letter-spacing 属性增加或减少字符间的空白（字符间距）。

该属性定义了在文本字符框之间插入多少空间。由于字符字形通常比其字符框要窄，指定长度值时，会调整字母之间通常的间隔。因此，normal 就相当于值为 0。

注释：允许使用负值，这会让字母之间挤得更紧。

默认值：	**normal**
继承性：	yes
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.letterSpacing="3px"
---	---

可能的值

值	描述
normal	默认。规定字符间没有额外的空间。
length	定义字符间的固定空间（允许使用负值）。
inherit	规定应该从父元素继承 letter-spacing 属性的值。

TIY 实例

规定字符间距（字母间隔）

本例演示如何增加或减少字符间距。

```
<html>

<head>
<style type="text/css">
h1 {letter-spacing: -0.5em}
h4 {letter-spacing: 20px}
</style>
</head>

<body>
<h1>This is header 1</h1>
<h4>This is header 4</h4>
</body>

</html>
```

相关页面

CSS 教程：CSS 文本

HTML DOM 参考手册：letterSpacing 属性

CSS line-height 属性

实例

设置以百分比计的行高：

```
p.small {line-height:90%}
p.big {line-height:200%}
```

（可以在页面底部查看更多实例）

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 line-height 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

line-height 属性设置行间的距离（行高）。

注释：不允许使用负值。

说明

该属性会影响行框的布局。在应用到一个块级元素时，它定义了该元素中基线之间的最小距离而不是最大距离。

line-height 与 font-size 的计算值之差（在 CSS 中成为“行间距”）分为两半，分别加到一个文本行内容的顶部和底部。可以包含这些内容的最小框就是行框。

原始数字值指定了一个缩放因子，后代元素会继承这个缩放因子而不是计算值。

默认值：	**normal**
继承性：	yes
版本：	CSS1
JavaScript 语法：	*object*.style.lineHeight="2"

可能的值

值	描述
normal	默认。设置合理的行间距。
number	设置数字，此数字会与当前的字体尺寸相乘来设置行间距。
length	设置固定的行间距。
%	基于当前字体尺寸的百分比行间距。
inherit	规定应该从父元素继承 line-height 属性的值。

TIY 实例

使用百分比设置行间距

本例演示如何使用百分比值来设置段落中的行间距。

```
<html>

<head>
<style type="text/css">
p.small {line-height: 90%}
p.big {line-height: 200%}
</style>
</head>

<body>

<p>
这是拥有标准行高的段落。
在大多数浏览器中默认行高大约是 110% 到 120%。
这是拥有标准行高的段落。
这是拥有标准行高的段落。
这是拥有标准行高的段落。
这是拥有标准行高的段落。
这是拥有标准行高的段落。
</p>

<p class="small">
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
</p>

<p class="big">
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
</p>

</body>
</html>
```

使用像素值设置行间距

本例演示如何使用像素值来设置段落中的行间距。

```
<html>

<head>
<style type="text/css">
p.small
  {
  line-height: 10px
  }
p.big
  {
  line-height: 30px
  }
</style>
</head>

<body>

<p>
这是拥有标准行高的段落。
在大多数浏览器中默认行高大约是 20px。
这是拥有标准行高的段落。
这是拥有标准行高的段落。
这是拥有标准行高的段落。
这是拥有标准行高的段落。
这是拥有标准行高的段落。
</p>

<p class="small">
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
</p>

<p class="big">
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
</p>

</body>
</html>
```

使用数值来设置行间距

本例演示如何使用一个数值来设置段落中的行间距。

```
<html>

<head>
<style type="text/css">
p.small
{
line-height: 0.5
}
p.big
{
line-height: 2
}
</style>
</head>

<body>

<p>
这是拥有标准行高的段落。
默认行高大约是 1。
这是拥有标准行高的段落。
这是拥有标准行高的段落。
这是拥有标准行高的段落。
这是拥有标准行高的段落。
这是拥有标准行高的段落。
</p>

<p class="small">
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
这个段落拥有更小的行高。
</p>

<p class="big">
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
这个段落拥有更大的行高。
</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 文本

HTML DOM 参考手册：lineHeight 属性

CSS text-align 属性

实例

设置 h1、h2、h3 元素的文本对齐方式：

```
h1 {text-align:center}
h2 {text-align:left}
h3 {text-align:right}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 text-align 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

text-align 属性规定元素中的文本的水平对齐方式。

该属性通过指定行框与哪个点对齐，从而设置块级元素内文本的水平对齐方式。通过允许用户代理调整行内容中字母和字之间的间隔，可以支持值 justify；不同用户代理可能会得到不同的结果。

默认值：	如果 **direction** 属性是 **ltr**，则默认值是 **left**；如果 **direction** 是 **rtl**，则为 **right**。
继承性：	yes
版本：	CSS1
JavaScript 语法：	*object*.style.textAlign="right"

可能的值

值	描述
left	把文本排列到左边。默认值：由浏览器决定。
right	把文本排列到右边。
center	把文本排列到中间。
justify	实现两端对齐文本效果。
inherit	规定应该从父元素继承 text-align 属性的值。

值 justify

最后一个水平对齐属性是 justify，它会带来自己的一些问题。

值 justify 可以使文本的两端都对齐。在两端对齐文本中，文本行的左右两端都放在父元素的内边界上。然后，调整单词和字母间的间隔，使各行的长度恰好相等。您也许已经注意到了，两端对齐文本在打印领域很常见。不过在 CSS 中，还需要多做些考虑。

要由用户代理（而不是 CSS）来确定两端对齐文本如何拉伸，以填满父元素左右边界之间的空间。例如，有些浏览器可能只在单词之间增加额外的空间，而另外一些浏览器可能会平均分布字母间的额外空间（不过 CSS 规范特别指出，如果 letter-spacing 属性指定为一个长度值，“用户代理不能进一步增加或减少字符间的空间”）。还有一些用户代理可能会减少某些行的空间，使文本挤得更紧密。所有这些做法都会影响元素的外观，甚至改变其高度，这取决于用户代理的对齐选择影响了多少文本行。

CSS 也没有指定应当如何处理连字符（注1）。大多数两端对齐文本都使用连字符将长单词分开放在两行上，从而缩小单词之间的间隔，改善文本行的外观。不过，由于 CSS 没有定义连字符行为，用户代理不太可能自动加连字符。因此，在 CSS 中，两端对齐文本看上去没有打印出来好看，特别是元素可能太窄，以至于每行只能放下几个单词。当然，使用窄设计元素是可以的，不过要当心相应的缺点。

注1：CSS 中没有说明如何处理连字符，因为不同的语言有不同的连字符规则。规范没有尝试去调和这样一些很可能不完备的规则，而是干脆不提这个问题。

TIY 实例

对齐文本

本例演示如何对齐文本。

```
<html>
<head>
<style type="text/css">
h1 {text-align: center}
h2 {text-align: left}
h3 {text-align: right}
</style>
</head>

<body>
<h1>这是标题 1</h1>
<h2>这是标题 2</h2>
<h3>这是标题 3</h3>
</body>

</html>
```

相关页面

CSS 教程：CSS 文本

HTML DOM 参考手册：textAlign 属性

CSS text-decoration 属性

实例

设置 h1、h2、h3、h4 元素的文本修饰：

```
h1 {text-decoration:overline}
h2 {text-decoration:line-through}
h3 {text-decoration:underline}
h4 {text-decoration:blink}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 text-decoration 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

注释：IE、Chrome 或 Safari 不支持 "blink" 属性值。

定义和用法

text-decoration 属性规定添加到文本的修饰。

注释：修饰的颜色由 "color" 属性设置。

说明

这个属性允许对文本设置某种效果，如加下划线。如果后代元素没有自己的装饰，祖先元素上设置的装饰会“延伸”到后代元素中。不要求用户代理支持 blink。

默认值：	none
继承性：	no
版本：	CSS1
JavaScript 语法：	*object*.style.textDecoration="overline"

可能的值

值	描述
none	默认。定义标准的文本。
underline	定义文本下的一条线。
overline	定义文本上的一条线。
line-through	定义穿过文本下的一条线。
blink	定义闪烁的文本。
inherit	规定应该从父元素继承 text-decoration 属性的值。

TIY 实例

修饰文本

本例演示如何向文本添加修饰。

```
<html>
<head>
<style type="text/css">
h1 {text-decoration: overline}
h2 {text-decoration: line-through}
h3 {text-decoration: underline}
h4 {text-decoration:blink}
a {text-decoration: none}
</style>
</head>

<body>
<h1>这是标题 1</h1>
<h2>这是标题 2</h2>
<h3>这是标题 3</h3>
<h4>这是标题 4</h4>
<p><a href="http://www.w3school.com.cn/index.html">这是一个链接</a></p>
</body>

</html>
```

相关页面

CSS 教程：CSS 文本

HTML DOM 参考手册：textDecoration 属性

CSS text-indent 属性

实例

将段落的第一行缩进 50 像素：

```
p
  {
  text-indent:50px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 text-indent 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义

text-indent 属性规定文本块中首行文本的缩进。

注释：允许使用负值。如果使用负值，那么首行会被缩进到左边。

注意：在 CSS 2.1 之前，text-indent 总是继承计算值，而不是声明值。

说明

用于定义块级元素中第一个内容行的缩进。这最常用于建立一个“标签页”效果。允许指定负值，这会产生一种“悬挂缩进”的效果。

默认值：	*not specified*
继承性：	yes
版本：	CSS1
JavaScript 语法：	*object*.style.textIndent="50px"

可能的值

值	描述
length	定义固定的缩进。默认值：0。
%	定义基于父元素宽度的百分比的缩进。
inherit	规定应该从父元素继承 text-indent 属性的值。

TIY 实例

缩进文本

本例演示如何缩进文本首行。

```
<html>
<head>
<style type="text/css">
p {text-indent: 1cm}
</style>
</head>

<body>
<p>
这是段落中的一些文本。
这是段落中的一些文本。
这是段落中的一些文本。
这是段落中的一些文本。
这是段落中的一些文本。
这是段落中的一些文本。
这是段落中的一些文本。
这是段落中的一些文本。
这是段落中的一些文本。
这是段落中的一些文本。
这是段落中的一些文本。
这是段落中的一些文本。
</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 文本

HTML DOM 参考手册：textIndent 属性

CSS text-transform 属性

实例

转换不同元素中的文本：

```
h1 {text-transform:uppercase}
h2 {text-transform:capitalize}
p {text-transform:lowercase}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 text-transform 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

text-transform 属性控制文本的大小写。

说明

这个属性会改变元素中的字母大小写，而不论源文档中文本的大小写。如果值为 capitalize，则要对某些字母大写，但是并没有明确定义如何确定哪些字母要大写，这取决于用户代理如何识别出各个“词”。

默认值：	none
继承性：	yes
版本：	CSS1
JavaScript 语法：	*object*.style.textTransform="uppercase"

提示和注释

注释：不同的用户代理可能会用不同的方法来确定单词从哪里开始，相应地确定哪些字母要大写。例如，文本 "w3-school" 可以用两种方式显示："W3-school" 和 "W3-School"。CSS 并没有规定哪一种是正确的，所以这两种都是可以的。

可能的值

值	描述
none	默认。定义带有小写字母和大写字母的标准的文本。
capitalize	文本中的每个单词以大写字母开头。
uppercase	定义仅有大写字母。
lowercase	定义无大写字母，仅有小写字母。
inherit	规定应该从父元素继承 text-transform 属性的值。

TIY 实例

控制文本中的字母

本例演示如何控制文本中的字母的大小写。

```
<html>

<head>
<style type="text/css">
  h1 {text-transform: uppercase}
  p.uppercase {text-transform: uppercase}
  p.lowercase {text-transform: lowercase}
  p.capitalize {text-transform: capitalize}
</style>
</head>

<body>
<h1>This Is An H1 Element</h1>
<p class="uppercase">This is some text in a paragraph.</p>
<p class="lowercase">This is some text in a paragraph.</p>
<p class="capitalize">This is some text in a paragraph.</p>
</body>

</html>
```

相关页面

CSS 教程：CSS 文本

HTML DOM 参考手册：textTransform 属性

CSS unicode-bidi 属性

定义

unicode-bidi 属性设置文本的方向。

继承性：Yes

说明

尽管 CSS 试图处理书写方向，但 Unicode 有一种更健壮的方式来处理方向性。利用属性 unicode-bidi，CSS 创作人员可以充分利用 Unicode 的某些功能。

可能的值

在这里，我们将简要引用 CSS2.1 规范中关于这些值的描述，这些描述很好地说明了各个值的实现。

normal

元素不会对双向算法打开附加的一层嵌套。对于行内元素，顺序的隐式重排会跨元素边界进行。

embed

如果是一个行内元素，这个值对于双向算法会打开附件的一层嵌套。这个嵌套层的方向由 direction 属性指定。会在元素内部隐式地完成顺序重排。这对应于在元素开始处增加一个 LRE（对于 direction:ltr ：U+202A）或 RLE（对于 direction:rtl ：U+202B），并在元素的最后增加一个 PDF（U+202C）。

bidi-override

这会为行内元素创建一个覆盖。对于块级元素，将为不在另一块中的行内后代创建一个覆盖。这说明，顺序重排在元素内部严格按照 direction 属性进行；忽略了双向算法的隐式部分。这对应于在元素开始处增加一个 LRO（对于 direction:ltr ：U+202D）或 RLO（对于 direction:rtl ：U+202E），并在元素最后增加一个 PDF（U+202C）。

CSS white-space 属性

实例

规定段落中的文本不进行换行：

```
p
  {
  white-space: nowrap
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 white-space 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义和用法

white-space 属性设置如何处理元素内的空白。

这个属性声明建立布局过程中如何处理元素中的空白符。值 pre-wrap 和 pre-line 是 CSS 2.1 中新增的。

默认值：	normal
继承性：	yes
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.whiteSpace="pre"
---	---

可能的值

值	描述
normal	默认。空白会被浏览器忽略。
pre	空白会被浏览器保留。其行为方式类似 HTML 中的 <pre> 标签。
nowrap	文本不会换行，文本会在在同一行上继续，直到遇到 标签为止。
pre-wrap	保留空白符序列，但是正常地进行换行。
pre-line	合并空白符序列，但是保留换行符。
inherit	规定应该从父元素继承 white-space 属性的值。

TIY 实例

在元素中禁止文本折行

本例演示如何禁止在元素中的文本折行。

```
<html>
<head>
<style type="text/css">
p
{
white-space: nowrap
}
</style>
</head>
<body>

<p>
这是一些文本。
这是一些文本。
这是一些文本。
这是一些文本。
这是一些文本。
这是一些文本。
这是一些文本。
这是一些文本。
这是一些文本。
这是一些文本。
这是一些文本。
这是一些文本。
</p>

</body>
</html>
```

相关页面

CSS 教程：CSS 文本

HTML DOM 参考手册：whiteSpace 属性

CSS word-spacing 属性

实例

规定段落中的字间距是 25 像素：

```
p
  {
  word-spacing:25px;
  }
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 word-spacing 属性。

注释：任何的版本的 Internet Explorer （包括 IE8）都不支持属性值 "inherit"。

定义

word-spacing 属性增加或减少单词间的空白（即字间隔）。

该属性定义元素中字之间插入多少空白符。针对这个属性，“字” 定义为由空白符包围的一个字符串。如果指定为长度值，会调整字之间的通常间隔；所以，normal 就等同于设置为 0。允许指定负长度值，这会让字之间挤得更紧。

注释：允许使用负值。

默认值：	normal
继承性：	yes
---	---
版本：	CSS1
---	---
JavaScript 语法：	*object*.style.wordSpacing="10px"
---	---

提示和注释

注释：CSS 把“字（word）”定义为任何非空白符字符组成的串，并由某种空白字符包围。这个定义没有实际的语义，它只是假设一个文档包含由一个或多个空白字符包围的字。支持 CSS 的用户代理不一定能确定一个给定语言中哪些是合法的字，而哪些不是。尽管这个定义没有多大价值，不过它意味着采用象形文字的语言或非罗马书写体往往无法指定字间隔。

提示：利用这个属性，可能会创建字间隔太宽的文档，所以，使用 word-spacing 时要小心。

可能的值

值	描述
normal	默认。定义单词间的标准空间。
length	定义单词间的固定空间。
inherit	规定应该从父元素继承 word-spacing 属性的值。

TIY 实例

增加或减少单词间距（字间隔）

本例演示如何增加段落中单词间的距离。

```
<html>
<head>
<style type="text/css">
p.spread {word-spacing: 30px;}
p.tight {word-spacing: -0.5em;}
</style>
</head>

<body>
<p class="spread">This is some text. This is some text.</p>
<p class="tight">This is some text. This is some text.</p>
</body>
</html>
```

相关页面

CSS 教程：CSS 文本

HTML DOM 参考手册：wordSpacing 属性

CSS3 hanging-punctuation 属性

实例

在 p 元素首行的开始边缘之外放置一个标点符号：

```
p
{
hanging-punctuation:first;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

目前主流浏览器都不支持 hanging-punctuation 属性。

定义和用法

hanging-punctuation 属性规定把标点符号放在文本整行的开头还是结尾的行框之外。

默认值：	none
继承性：	yes
版本：	CSS3
JavaScript 语法：	*object*.style.hangingPunctuation="first"

语法

```
hanging-punctuation: none|first|last|allow-end|force-end;
```

值	描述
none	不在文本整行的开头还是结尾的行框之外放置标签符号。
first	标点附着在首行开始边缘之外。
last	标点附着在首行结尾边缘之外。
allow-end	
force-end	

CSS3 punctuation-trim 属性

实例

修剪 p 元素中每行开头的开启标点：

```
p
{
punctuation-trim:start;
}
```

浏览器支持

目前主流浏览器都不支持 punctuation-trim 属性。

定义和用法

punctuation-trim 属性规定如果标点位于行开头或结尾处，或者临近另一个全角标点符号，是否对标点符号进行修剪。

默认值：	none
继承性：	yes
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.punctuationTrim="start"
---	---

语法

```
punctuation-trim: none|start|end|allow-end|adjacent;
```

值	描述
none	不修剪开启或闭合标点符号。
start	修剪每行结尾的开启标点符号。
end	修剪每行结尾的闭合标点符号。
allow-end	
adjacent	

CSS3 text-emphasis 属性

浏览器支持

目前主流浏览器都不支持 text-emphasis 属性。

定义和用法

text-emphasis 属性是简写属性，用于在一个声明中设置 text-emphasis-style 和 text-emphasis-color。

提示：Adobe 的 "Kenten Generic OpenType Font" 是一个适合重点标记的字体，它专门为重点标记设计。

默认值：	**none**
继承性：	yes
版本：	CSS3
JavaScript 语法：	*object*.style.textEmphasis="filled blue"

语法

```
text-emphasis: _text-emphasis-style_ _text-emphasis-color_;
```

值	描述
text-emphasis-style	向元素的文本应用重点标记。
text-emphasis-color	定义重点标记的前景色。

CSS3 text-justify 属性

实例

齐行改变单词间的间隔：

```
div
{
text-align:justify;
text-justify:inter-word;
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

只有 Internet Explorer 支持 text-justify 属性。

定义和用法

text-justify 属性规定当 text-align 被设置为 text-align 时的齐行方法。

该属性规定如何对齐行文本进行对齐和分隔。

默认值：	auto
继承性：	yes
版本：	CSS3
JavaScript 语法：	*object*.style.textJustify="inter-word"

语法

```
text-justify: auto|inter-word|inter-ideograph|inter-cluster|distribute|kashida|trim;
```

值	描述	测试
auto	浏览器决定齐行算法。	测试
none	禁用齐行。	测试
inter-word	增加/减少单词间的间隔。	测试
inter-ideograph	用表意文本来排齐内容。	测试
inter-cluster	只对不包含内部单词间隔的内容（比如亚洲语系）进行排齐。	测试
distribute	类似报纸版面，除了在东亚语系中最后一行是不齐行的。	测试
kashida	通过拉伸字符来排齐内容。	测试

CSS3 text-outline 属性

实例

设置 text-outline：

```
p.test
{
text-outline: 2px 2px #ff0000;
}
```

浏览器支持

所有主流浏览器都不支持 text-outline 属性。

定义和用法

text-outline 属性规定文本轮廓。

默认值：	none
继承性：	yes
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.textOutline="2px 2px #ff0000"
---	---

语法

```
text-outline: _thickness_ _blur_ _color_;
```

值	描述
thickness	必需。轮廓的粗细。
blur	可选。轮廓的模糊半径。
color	必需。轮廓的颜色。参阅 CSS 颜色值。

CSS3 text-overflow 属性

实例

使用 text-overflow 属性：

```
div.test
{
text-overflow:ellipsis;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 text-overflow 属性。

定义和用法

text-overflow 属性规定当文本溢出包含元素时发生的事情。

默认值：	clip
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.textOverflow="ellipsis"

语法

```
text-overflow: clip|ellipsis|_string_;
```

值	描述	测试
clip	修剪文本。	测试
ellipsis	显示省略符号来代表被修剪的文本。	测试
string	使用给定的字符串来代表被修剪的文本。	

亲自试一试 - 实例

带有 hover 效果的 Text-overflow

本例演示当光标浮动到元素上时如何显示全部文本。

```
<!DOCTYPE html>
<html>
<head>
<style>
div.test
{
white-space:nowrap;
width:12em;
overflow:hidden;
border:1px solid #000000;
}

div.test:hover
{
text-overflow:inherit;
overflow:visible;
}
</style>

</head>

<body>

<p>如果您把光标移动到下面两个 div 上，就能够看到全部文本。</p>

<p>这个 div 使用 "text-overflow:ellipsis" ：</p>

<div class="test" style="text-overflow:ellipsis;">This is some long text that will not fi

<p>这个 div 使用 "text-overflow:clip"：</p>

<div class="test" style="text-overflow:clip;">This is some long text that will not fit in

</body>
</html>
```

相关页面

CSS3 教程：CSS3 文本效果

CSS3 text-shadow 属性

实例

基础的文本阴影效果：

```
h1
{
text-shadow: 5px 5px 5px #FF0000;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 text-shadow 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 text-shadow 属性。

定义和用法

text-shadow 属性向文本设置阴影。

默认值：	none
继承性：	yes
版本：	CSS3
JavaScript 语法：	*object*.style.textShadow="2px 2px #ff0000"

语法

```
text-shadow: _h-shadow_ _v-shadow_ _blur_ _color_;
```

注释：text-shadow 属性向文本添加一个或多个阴影。该属性是逗号分隔的阴影列表，每个阴影有两个或三个长度值和一个可选的颜色值进行规定。省略的长度是 0。

值	描述	测试
h-shadow	必需。水平阴影的位置。允许负值。	测试
v-shadow	必需。垂直阴影的位置。允许负值。	测试
blur	可选。模糊的距离。	测试
color	可选。阴影的颜色。参阅 CSS 颜色值。	测试

亲自试一试 - 实例

带有模糊效果的文本阴影

该例演示带有模糊效果的文本阴影。

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {text-shadow:2px 2px 8px #FF0000;}
</style>
</head>
<body>

<h1>模糊效果的文本阴影！</h1>

</body>
</html>
```

白色文本上的阴影

本例演示白色文本上的文本阴影。

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
color:white;
text-shadow:2px 2px 4px #000000;
}
</style>
</head>
<body>

<h1>白色文本的阴影效果！</h1>

</body>
</html>
```

霓虹灯效果的文本阴影

本例演示带有霓虹灯效果的文本阴影。

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
text-shadow:0 0 3px #FF0000;
}
</style>
</head>
<body>

<h1>霓虹灯效果的文本阴影！</h1>

</body>
</html>
```

相关页面

CSS3 教程：CSS3 文本效果

CSS3 text-wrap 属性

实例

不允许换行：

```
p.test {text-wrap:none;}
```

浏览器支持

目前主流浏览器都不支持 text-wrap 属性。

定义和用法

text-wrap 属性规定文本的换行（折行）规则。

默认值：	normal
继承性：	yes
版本：	CSS3
JavaScript 语法：	*object*.style.textWrap="none"

语法

```
text-wrap: normal|none|unrestricted|suppress;
```

值	描述
normal	只在允许的换行点进行换行。
none	不换行。元素无法容纳的文本会溢出。
unrestricted	在任意两个字符间换行。
suppress	压缩元素中的换行。浏览器只在行中没有其他有效换行点时进行换行。

CSS3 word-break 属性

实例

在恰当的断字点进行换行：

```
p.test {word-break:hyphenate;}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 word-break 属性。

定义和用法

word-break 属性规定自动换行的处理方法。

提示：通过使用 word-break 属性，可以让浏览器实现在任意位置的换行。

默认值：	normal
继承性：	yes
版本：	CSS3
JavaScript 语法：	*object*.style.wordBreak="keep-all"

语法

```
word-break: normal|break-all|keep-all;
```

值	描述
normal	使用浏览器默认的换行规则。
break-all	允许在单词内换行。
keep-all	只能在半角空格或连字符处换行。

CSS3 word-wrap 属性

实例

允许长单词换行到下一行：

```
p.test {word-wrap:break-word;}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 word-wrap 属性。

定义和用法

word-wrap 属性允许长单词或 URL 地址换行到下一行。

默认值：	normal
继承性：	yes
版本：	CSS3
JavaScript 语法：	*object*.style.wordWrap="break-word"

语法

```
word-wrap: normal|break-word;
```

值	描述
normal	只在允许的断字点换行（浏览器保持默认处理）。
break-word	在长单词或 URL 地址内部进行换行。

2D/3D 转换属性（Transform）

属性	描述	CSS
transform	向元素应用 2D 或 3D 转换。	3
transform-origin	允许你改变被转换元素的位置。	3
transform-style	规定被嵌套元素如何在 3D 空间中显示。	3
perspective	规定 3D 元素的透视效果。	3
perspective-origin	规定 3D 元素的底部位置。	3
backface-visibility	定义元素在不面对屏幕时是否可见。	3

CSS3 transform 属性

实例

旋转 div 元素：

```
div
{
transform:rotate(7deg);
-ms-transform:rotate(7deg);     /* IE 9 */
-moz-transform:rotate(7deg);     /* Firefox */
-webkit-transform:rotate(7deg); /* Safari 和 Chrome */
-o-transform:rotate(7deg);     /* Opera */
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox、Opera 支持 transform 属性。

Internet Explorer 9 支持替代的 -ms-transform 属性（仅适用于 2D 转换）。

Safari 和 Chrome 支持替代的 -webkit-transform 属性（3D 和 2D 转换）。

Opera 只支持 2D 转换。

定义和用法

transform 属性向元素应用 2D 或 3D 转换。该属性允许我们对元素进行旋转、缩放、移动或倾斜。

为了更好地理解 transform 属性，请查看这个演示。

默认值：	**none**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.transform="rotate(7deg)"

语法

```
transform: none|_transform-functions_;
```

值	描述	测试
none	定义不进行转换。	测试
matrix(*n*,*n*,*n*,*n*,*n*,*n*)	定义 2D 转换，使用六个值的矩阵。	测试
matrix3d(*n*,*n*,*n*,*n*,*n*,*n*,*n*,*n*,*n*,*n*,*n*,*n*,*n*,*n*,*n*,*n*)	定义 3D 转换，使用 16 个值的 4x4 矩阵。	
translate(*x*,*y*)	定义 2D 转换。	测试
translate3d(*x*,*y*,*z*)	定义 3D 转换。	
translateX(*x*)	定义转换，只是用 X 轴的值。	测试
translateY(*y*)	定义转换，只是用 Y 轴的值。	测试
translateZ(*z*)	定义 3D 转换，只是用 Z 轴的值。	
scale(*x*,*y*)	定义 2D 缩放转换。	测试
scale3d(*x*,*y*,*z*)	定义 3D 缩放转换。	
scaleX(*x*)	通过设置 X 轴的值来定义缩放转换。	测试
scaleY(*y*)	通过设置 Y 轴的值来定义缩放转换。	测试
	通过设置 Z 轴的值来定义 3D 缩放转	

	换。	
rotate(*angle*)	定义 2D 旋转，在参数中规定角度。	测试
rotate3d(*x*,*y*,*z*,*angle*)	定义 3D 旋转。	
rotateX(*angle*)	定义沿着 X 轴的 3D 旋转。	测试
rotateY(*angle*)	定义沿着 Y 轴的 3D 旋转。	测试
rotateZ(*angle*)	定义沿着 Z 轴的 3D 旋转。	测试
skew(*x-angle*,*y-angle*)	定义沿着 X 和 Y 轴的 2D 倾斜转换。	测试
skewX(*angle*)	定义沿着 X 轴的 2D 倾斜转换。	测试
skewY(*angle*)	定义沿着 Y 轴的 2D 倾斜转换。	测试
perspective(*n*)	为 3D 转换元素定义透视视图。	测试

亲自试一试 - 实例

扔到桌子上面的图片

本例演示如何创建“宝丽来”图片，并旋转图片。

```
<!DOCTYPE html>
<html>
<head>
<style>
body
{
margin:30px;
background-color:#E9E9E9;
}

div.polaroid
{
width:294px;
padding:10px 10px 20px 10px;
border:1px solid #BFBFBF;
background-color:white;
/* Add box-shadow */
box-shadow:2px 2px 3px #aaaaaa;
}

div.rotate_left
{
float:left;
-ms-transform:rotate(7deg); /* IE 9 */
-moz-transform:rotate(7deg); /* Firefox */
-webkit-transform:rotate(7deg); /* Safari and Chrome */
-o-transform:rotate(7deg); /* Opera */
transform:rotate(7deg);
}

div.rotate_right
{
float:left;
-ms-transform:rotate(-8deg); /* IE 9 */
-moz-transform:rotate(-8deg); /* Firefox */
-webkit-transform:rotate(-8deg); /* Safari and Chrome */
-o-transform:rotate(-8deg); /* Opera */
transform:rotate(-8deg);
}
</style>
</head>
<body>

<div class="polaroid rotate_left">
<img src="/i/ballade_dream.jpg" alt="郁金香" width="284" height="213" />
<p class="caption">上海鲜花港的郁金香，花名：Ballade Dream。</p>
</div>

<div class="polaroid rotate_right">
<img src="/i/china_pavilion.jpg" alt="世博中国馆" width="284" height="213" />
<p class="caption">2010年上海世博会，中国馆。</p>
</div>

</body>
</html>
```

相关页面

CSS3 教程：CSS3 2D 转换

CSS3 教程：CSS3 3D 转换

CSS3 transform-origin 属性

实例

设置旋转元素的基点位置：

```
div
{
transform: rotate(45deg);
transform-origin:20% 40%;

-ms-transform: rotate(45deg);         /* IE 9 */
-ms-transform-origin:20% 40%;         /* IE 9 */

-webkit-transform: rotate(45deg);    /* Safari 和 Chrome */
-webkit-transform-origin:20% 40%;    /* Safari 和 Chrome */

-moz-transform: rotate(45deg);        /* Firefox */
-moz-transform-origin:20% 40%;        /* Firefox */

-o-transform: rotate(45deg);        /* Opera */
-o-transform-origin:20% 40%;        /* Opera */
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox、Opera 支持 transform-origin 属性。

Internet Explorer 9 支持替代的 -ms-transform-origin 属性（仅适用于 2D 转换）。

Safari 和 Chrome 支持替代的 -webkit-transform-origin 属性（3D 和 2D 转换）。

Opera 只支持 2D 转换。

定义和用法

transform-origin 属性允许您改变被转换元素的位置。

2D 转换元素能够改变元素 x 和 y 轴。3D 转换元素还能改变其 Z 轴。

为了更好地理解 transform-origin 属性，请查看这个演示。

Safari/Chrome 用户：为了更好地理解 transform-origin 属性用于 3D 转换的情况，请查看这个演示。

注释：该属性必须与 transform 属性一同使用。

为了更好地理解 transform 属性，请查看这个演示。

默认值：	**50% 50% 0**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.transformOrigin="20% 40%"

语法

```
transform-origin: _x-axis_ _y-axis_ _z-axis_;
```

值	描述
x-axis	

定义视图被置于 X 轴的何处。可能的值：

- left
- center
- right
- *length*
- *%*

  | | y-axis |

定义视图被置于 Y 轴的何处。可能的值：

- top
- center
- bottom
- *length*
- *%*

  | | z-axis |

定义视图被置于 Z 轴的何处。可能的值：

- *length*

  l

相关页面

CSS3 教程：CSS3 2D 转换

CSS3 教程：CSS3 3D 转换

CSS3 transform-style 属性

实例

使被转换的子元素保留其 3D 转换：

```
div
{
transform: rotateY(60deg);
transform-style: preserve-3d;
-webkit-transform: rotateY(60deg);    /* Safari 和 Chrome */
-webkit-transform-style: preserve-3d;    /* Safari 和 Chrome */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Firefox 支持 transform-style 属性。

Chrome、Safari 和 Opera 支持替代的 -webkit-transform-style 属性。

定义和用法

transform-style 属性规定如何在 3D 空间中呈现被嵌套的元素。

注释：该属性必须与 transform 属性一同使用。

默认值：	**flat**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.transformStyle="preserve-3d"

语法

```
transform-style: flat|preserve-3d;
```

值	描述
flat	子元素将不保留其 3D 位置。
preserve-3d	子元素将保留其 3D 位置。

CSS3 perspective 属性

实例

设置元素被查看位置的视图：

```
div
{
perspective: 500;
-webkit-perspective: 500; /* Safari 和 Chrome */
}
```

浏览器支持

目前浏览器都不支持 perspective 属性。

Chrome 和 Safari 支持替代的 -webkit-perspective 属性。

定义和用法

perspective 属性定义 3D 元素距视图的距离，以像素计。该属性允许您改变 3D 元素查看 3D 元素的视图。

当为元素定义 perspective 属性时，其子元素会获得透视效果，而不是元素本身。

注释：perspective 属性只影响 3D 转换元素。

提示：请与 perspective-origin 属性一同使用该属性，这样您就能够改变 3D 元素的底部位置。

默认值：	**none**
继承性：	yes
版本：	CSS3
JavaScript 语法：	*object*.style.perspective=500

语法

```
perspective: _number_|none;
```

值	描述
number	元素距离视图的距离，以像素计。
none	默认值。与 0 相同。不设置透视。

CSS3 perspective-origin 属性

实例

设置 3D 元素的基点位置：

```
div
{
perspective:150;
perspective-origin: 10% 10%;
-webkit-perspective:150;    /* Safari 和 Chrome */
-webkit-perspective-origin: 10% 10%;    /* Safari 和 Chrome */
}
```

浏览器支持

目前浏览器都不支持 perspective-origin 属性。

Chrome 和 Safari 支持替代的 -webkit-perspecitve-origin 属性。

定义和用法

perspective-origin 属性定义 3D 元素所基于的 X 轴和 Y 轴。该属性允许您改变 3D 元素的底部位置。

当为元素定义 perspective-origin 属性时，其子元素会获得透视效果，而不是元素本身。

注释：该属性必须与 perspective 属性一同使用，而且只影响 3D 转换元素。

默认值：	**50% 50%**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.perspectiveOrigin="10% 10%"

语法

```
perspective-origin: _x-axis_ _y-axis_;
```

值	描述
x-axis	

定义该视图在 x 轴上的位置。默认值：50%。

可能的值：

- left
- center
- right
- *length*
- *%*

  | | *y-axis* |

定义该视图在 y 轴上的位置。默认值：50%。

可能的值：

- top
- center
- bottom
- *length*
- *%*

  |

CSS3 backface-visibility 属性

实例

隐藏被旋转的 div 元素的背面：

```
div
{
backface-visibility:hidden;
-webkit-backface-visibility:hidden;    /* Chrome 和 Safari */
-moz-backface-visibility:hidden;     /* Firefox */
-ms-backface-visibility:hidden;     /* Internet Explorer */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

只有 Internet Explorer 10+ 和 Firefox 支持 backface-visibility 属性。

Opera 15+、Safari 和 Chrome 支持替代的 -webkit-backface-visibility 属性。

定义和用法

backface-visibility 属性定义当元素不面向屏幕时是否可见。

如果在旋转元素不希望看到其背面时，该属性很有用。

默认值：	visible
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.backfaceVisibility="hidden"

语法

```
backface-visibility: visible|hidden;
```

值	描述
visible	背面是可见的。
hidden	背面是不可见的。

过渡属性（Transition）

属性	描述	CSS
transition	简写属性，用于在一个属性中设置四个过渡属性。	3
transition-property	规定应用过渡的 CSS 属性的名称。	3
transition-duration	定义过渡效果花费的时间。	3
transition-timing-function	规定过渡效果的时间曲线。	3
transition-delay	规定过渡效果何时开始。	3

CSS3 transition 属性

实例

把鼠标指针放到 div 元素上，其宽度会从 100px 逐渐变为 300px：

```
div
{
width:100px;
transition: width 2s;
-moz-transition: width 2s; /* Firefox 4 */
-webkit-transition: width 2s; /* Safari 和 Chrome */
-o-transition: width 2s; /* Opera */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox、Opera 和 Chrome 支持 transition 属性。

Safari 支持替代的 -webkit-transition 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 transition 属性。

定义和用法

transition 属性是一个简写属性，用于设置四个过渡属性：

- transition-property
- transition-duration
- transition-timing-function
- transition-delay

注释：请始终设置 transition-duration 属性，否则时长为 0，就不会产生过渡效果。

默认值：	all 0 ease 0
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.transition="width 2s"

语法

```
transition: _property_ _duration_ _timing-function_ _delay_;
```

值	描述
transition-property	规定设置过渡效果的 CSS 属性的名称。
transition-duration	规定完成过渡效果需要多少秒或毫秒。
transition-timing-function	规定速度效果的速度曲线。
transition-delay	定义过渡效果何时开始。

CSS3 transition-property 属性

实例

把鼠标指针放到 div 元素上，会产生带有平滑改变元素宽度的过渡效果：

```
div
{
transition-property:width;
-moz-transition-property: width; /* Firefox 4 */
-webkit-transition-property:width; /* Safari 和 Chrome */
-o-transition-property:width; /* Opera */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox、Opera 和 Chrome 支持 transition-property 属性。

Safari 支持替代的 -webkit-transition-property 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 transition-property 属性。

定义和用法

transition-property 属性规定应用过渡效果的 CSS 属性的名称。（当指定的 CSS 属性改变时，过渡效果将开始）。

提示：过渡效果通常在用户将鼠标指针浮动到元素上时发生。

注释：请始终设置 transition-duration 属性，否则时长为 0，就不会产生过渡效果。

默认值：	all
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.transitionProperty="width,height"
---	---

语法

```
transition-property: none|all|_property_;
```

值	描述
none	没有属性会获得过渡效果。
all	所有属性都将获得过渡效果。
property	定义应用过渡效果的 CSS 属性名称列表，列表以逗号分隔。

CSS3 transition-duration 属性

实例

让过渡效果持续 5 秒：

```
div
{
transition-duration: 5s;
-moz-transition-duration: 5s; /* Firefox 4 */
-webkit-transition-duration: 5s; /* Safari 和 Chrome */
-o-transition-duration: 5s; /* Opera */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox、Opera 和 Chrome 支持 transition-duration 属性。

Safari 支持替代的 -webkit-transition-duration 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 transition-duration 属性。

定义和用法

transition-duration 属性规定完成过渡效果需要花费的时间（以秒或毫秒计）。

默认值：	0
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.transitionDuration="5s"

语法

```
transition-duration: _time_;
```

值	描述
time	

规定完成过渡效果需要花费的时间（以秒或毫秒计）。

默认值是 0，意味着不会有效果。

|

CSS3 transition-timing-function 属性

实例

以相同的速度从开始到结束的过渡效果：

```
div
{
transition-timing-function: linear;
-moz-transition-timing-function: linear; /* Firefox 4 */
-webkit-transition-timing-function: linear; /* Safari 和 Chrome */
-o-transition-timing-function: linear; /* Opera */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox、Opera 和 Chrome 支持 transition-timing-function 属性。

Safari 支持替代的 -webkit-transition-timing-function 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 transition-timing-function 属性。

定义和用法

transition-timing-function 属性规定过渡效果的速度曲线。

该属性允许过渡效果随着时间来改变其速度。

默认值：	ease
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.transitionTimingFunction="linear"
---	---

语法

```
transition-timing-function: linear|ease|ease-in|ease-out|ease-in-out|cubic-
bezier(_n_,_n_,_n_,_n_);
```

值	描述
linear	规定以相同速度开始至结束的过渡效果（等于 cubic-bezier(0,0,1,1)）。
ease	规定慢速开始，然后变快，然后慢速结束的过渡效果（cubic-bezier(0.25,0.1,0.25,1)）。
ease-in	规定以慢速开始的过渡效果（等于 cubic-bezier(0.42,0,1,1)）。
ease-out	规定以慢速结束的过渡效果（等于 cubic-bezier(0,0,0.58,1)）。
ease-in-out	规定以慢速开始和结束的过渡效果（等于 cubic-bezier(0.42,0,0.58,1)）。
cubic-bezier(*n*,*n*,*n*,*n*)	在 cubic-bezier 函数中定义自己的值。可能的值是 0 至 1 之间的数值。

提示：请在实例中测试不同的值，这样可以更好地理解它们的工作原理。

亲自试一试 - 实例

实例 1

为了更好地理解不同的函数值，请看下面带有五个不同值的五个不同的 div 元素：

```
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}
/* Firefox 4: */
#div1 {-moz-transition-timing-function: linear;}
#div2 {-moz-transition-timing-function: ease;}
#div3 {-moz-transition-timing-function: ease-in;}
#div4 {-moz-transition-timing-function: ease-out;}
#div5 {-moz-transition-timing-function: ease-in-out;}
/* Safari and Chrome: */
#div1 {-webkit-transition-timing-function: linear;}
#div2 {-webkit-transition-timing-function: ease;}
#div3 {-webkit-transition-timing-function: ease-in;}
#div4 {-webkit-transition-timing-function: ease-out;}
#div5 {-webkit-transition-timing-function: ease-in-out;}
/* Opera: */
#div1 {-o-transition-timing-function: linear;}
#div2 {-o-transition-timing-function: ease;}
#div3 {-o-transition-timing-function: ease-in;}
#div4 {-o-transition-timing-function: ease-out;}
#div5 {-o-transition-timing-function: ease-in-out;}
```

实例 2

与上例相同，但通过 cubic-bezier 来规定速度曲线：

```
#div1 {transition-timing-function: cubic-bezier(0,0,1,1;}
#div2 {transition-timing-function: cubic-bezier(0.25,0.1,0.25,1);}
#div3 {transition-timing-function: cubic-bezier(0.42,0,1,1);}
#div4 {transition-timing-function: cubic-bezier(0,0,0.58,1);}
#div5 {transition-timing-function: cubic-bezier(0.42,0,0.58,1);}
/* Firefox 4: */
#div1 {-moz-transition-timing-function: cubic-bezier(0,0,0.25,1);}
#div2 {-moz-transition-timing-function: cubic-bezier(0.25,0.1,0.25,1);}
#div3 {-moz-transition-timing-function: cubic-bezier(0.42,0,1,1);}
#div4 {-moz-transition-timing-function: cubic-bezier(0,0,0.58,1);}
#div5 {-moz-transition-timing-function: cubic-bezier(0.42,0,0.58,1);}
/* Safari and Chrome: */
#div1 {-webkit-transition-timing-function: cubic-bezier(0,0,1,1;}
#div2 {-webkit-transition-timing-function: cubic-bezier(0.25,0.1,0.25,1);}
#div3 {-webkit-transition-timing-function: cubic-bezier(0.42,0,1,1);}
#div4 {-webkit-transition-timing-function: cubic-bezier(0,0,0.58,1);}
#div5 {-webkit-transition-timing-function: cubic-bezier(0.42,0,0.58,1);}
/* Opera: */
#div1 {-o-transition-timing-function: cubic-bezier(0,0,1,1;}
#div2 {-o-transition-timing-function: cubic-bezier(0.25,0.1,0.25,1);}
#div3 {-o-transition-timing-function: cubic-bezier(0.42,0,1,1);}
#div4 {-o-transition-timing-function: cubic-bezier(0,0,0.58,1);}
#div5 {-o-transition-timing-function: cubic-bezier(0.42,0,0.58,1);}
```

CSS3 transition-delay 属性

实例

在过渡效果开始前等待 2 秒：

```
div
{
transition-delay: 2s;
-moz-transition-delay: 2s; /* Firefox 4 */
-webkit-transition-delay: 2s; /* Safari 和 Chrome */
-o-transition-delay: 2s; /* Opera */
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10、Firefox、Opera 和 Chrome 支持 transition-delay 属性。

Safari 支持替代的 -webkit-transition-delay 属性。

注释：Internet Explorer 9 以及更早版本的浏览器不支持 transition-delay 属性。

定义和用法

transition-delay 属性规定过渡效果何时开始。

transition-delay 值以秒或毫秒计。

默认值：	0
继承性：	no
---	---
版本：	CSS3
---	---
JavaScript 语法：	*object*.style.transitionDelay="2s"
---	---

语法

```
transition-delay: _time_;
```

值	描述
time	规定在过渡效果开始之前需要等待的时间，以秒或毫秒计。

用户界面属性（User-interface）

属性	描述	CSS
appearance	允许您将元素设置为标准用户界面元素的外观	3
box-sizing	允许您以确切的方式定义适应某个区域的具体内容。	3
icon	为创作者提供使用图标化等价物来设置元素样式的能力。	3
nav-down	规定在使用 arrow-down 导航键时向何处导航。	3
nav-index	设置元素的 tab 键控制次序。	3
nav-left	规定在使用 arrow-left 导航键时向何处导航。	3
nav-right	规定在使用 arrow-right 导航键时向何处导航。	3
nav-up	规定在使用 arrow-up 导航键时向何处导航。	3
outline-offset	对轮廓进行偏移，并在超出边框边缘的位置绘制轮廓。	3
resize	规定是否可由用户对元素的尺寸进行调整。	3

CSS3 appearance 属性

实例

使 div 元素看上去像一个按钮：

```
div
{
appearance:button;
-moz-appearance:button; /* Firefox */
-webkit-appearance:button; /* Safari 和 Chrome */
}
```

页面底部有更多实例。

浏览器支持

所有主流浏览器都不支持 appearance 属性。

Firefox 支持替代的 -moz-appearance 属性。

Safari 和 Chrome 支持替代的 -webkit-appearance 属性。

定义和用法

appearance 属性允许您使元素看上去像标准的用户界面元素。

默认值：	normal
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.appearance="button"

语法

```
appearance: normal|icon|window|button|menu|field;
```

值	描述
normal	将元素呈现为常规元素。
icon	将元素呈现为图标（小图片）。
window	将元素呈现为视口。
button	将元素呈现为按钮。
menu	将元素呈现为一套供用户选择的选项。
field	将元素呈现为输入字段。

CSS3 box-sizing 属性

实例

规定两个并排的带边框的框：

```
div
{
box-sizing:border-box;
-moz-box-sizing:border-box; /* Firefox */
-webkit-box-sizing:border-box; /* Safari */
width:50%;
float:left;
}
```

页面底部有更多实例。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer、Opera 以及 Chrome 支持 box-sizing 属性。

Firefox 支持替代的 -moz-box-sizing 属性。

定义和用法

box-sizing 属性允许您以特定的方式定义匹配某个区域的特定元素。

例如，假如您需要并排放置两个带边框的框，可通过将 box-sizing 设置为 "border-box"。这可令浏览器呈现出带有指定宽度和高度的框，并把边框和内边距放入框中。

默认值：	content-box
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.boxSizing="border-box"

语法

```
box-sizing: content-box|border-box|inherit;
```

值	描述
content-box	

这是由 CSS2.1 规定的宽度高度行为。

宽度和高度分别应用到元素的内容框。

在宽度和高度之外绘制元素的内边距和边框。

| | border-box |

为元素设定的宽度和高度决定了元素的边框盒。

就是说，为元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制。

通过从已设定的宽度和高度分别减去边框和内边距才能得到内容的宽度和高度。

| | inherit | 规定应从父元素继承 box-sizing 属性的值。 |

相关页面

CSS3 教程：CSS3 用户界面

CSS3 icon 属性

实例

将图像元素设置为图标化的等价物：

```
img
{
content:icon;
icon:url(imgicon.png);
}
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

目前没有浏览器支持 icon 属性。

定义和用法

icon 属性为创作者提供了将元素设置为图标等价物的能力。

注释：除非 "content" 属性的值被设置为 "icon"，否则元素的图标不会被使用。

默认值：	auto
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.icon="url(image.png)"

语法

```
icon: auto|_URL_|inherit;
```

值	描述
auto	使用由浏览器提供的默认通用图标。
URL	引用列表中的一个或多个图标，列表用逗号分隔。
inherit	规定应从元素继承 icon 属性的值。

CSS3 nav-down 属性

实例

规定在使用方向键时向何处导航：

```
button#b1
{
top:20%;left:25%;
nav-index:1;
nav-right:#b2;nav-left:#b4;
nav-down:#b2;nav-up:#b4;
}

button#b2
{
top:40%;left:50%;
nav-index:2;
nav-right:#b3;nav-left:#b1;
nav-down:#b3;nav-up:#b1;
}

button#b3
{
top:70%;left:25%;
nav-index:3;
nav-right:#b4;nav-left:#b2;
nav-down:#b4;nav-up:#b2;
}

button#b4
{
top:40%;left:0%;
nav-index:4;
nav-right:#b1;nav-left:#b3;
nav-down:#b1;nav-up:#b3;
}
```

浏览器支持

目前只有 Opera 支持 nav-down 属性。

定义和用法

nav-down 属性规定当使用 nav-down 导航键时，向何处进行导航。

默认值：	**auto**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.navDown="#div2"

语法

```
nav-down: auto|_id_|_target-name_|inherit;
```

值	描述
auto	浏览器决定导航到哪个元素。
id	规定被导航元素的 id。
target-name	规定被导航的目标框架。
inherit	规定应从父元素继承 nav-down 属性的值。

CSS3 nav-index 属性

实例

规定在使用方向键时向何处导航：

```
button#b1
{
top:20%;left:25%;
nav-index:1;
nav-right:#b2;nav-left:#b4;
nav-down:#b2;nav-up:#b4;
}

button#b2
{
top:40%;left:50%;
nav-index:2;
nav-right:#b3;nav-left:#b1;
nav-down:#b3;nav-up:#b1;
}

button#b3
{
top:70%;left:25%;
nav-index:3;
nav-right:#b4;nav-left:#b2;
nav-down:#b4;nav-up:#b2;
}

button#b4
{
top:40%;left:0%;
nav-index:4;
nav-right:#b1;nav-left:#b3;
nav-down:#b1;nav-up:#b3;
}
```

浏览器支持

目前只有 Opera 支持 nav-index 属性。

定义和用法

nav-index 属性规定元素的连续导航次序 ("tabbing order")。

默认值：	**auto**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.navIndex=2

语法

```
nav-index: auto|_number_|inherit;
```

值	描述
auto	由浏览器分配元素的导航键控制次序。
number	指示元素的导航键控制次序。1 代表第一个。
inherit	规定应从父元素继承 nav-index 属性的值。

CSS3 nav-left 属性

实例

规定在使用方向键时向何处导航：

```
button#b1
{
top:20%;left:25%;
nav-index:1;
nav-right:#b2;nav-left:#b4;
nav-down:#b2;nav-up:#b4;
}

button#b2
{
top:40%;left:50%;
nav-index:2;
nav-right:#b3;nav-left:#b1;
nav-down:#b3;nav-up:#b1;
}

button#b3
{
top:70%;left:25%;
nav-index:3;
nav-right:#b4;nav-left:#b2;
nav-down:#b4;nav-up:#b2;
}

button#b4
{
top:40%;left:0%;
nav-index:4;
nav-right:#b1;nav-left:#b3;
nav-down:#b1;nav-up:#b3;
}
```

浏览器支持

目前只有 Opera 支持 nav-left 属性。

定义和用法

nav-left 属性规定当使用 nav-left 导航键时，向何处进行导航。

默认值：	**auto**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.navLeft="#div2"

语法

```
nav-left: auto|_id_|_target-name_|inherit;
```

值	描述
auto	浏览器决定导航到哪个元素。
id	规定被导航元素的 id。
target-name	规定被导航的目标框架。
inherit	规定应从父元素继承 nav-left 属性的值。

CSS3 nav-right 属性

实例

规定在使用方向键时向何处导航：

```
button#b1
{
top:20%;left:25%;
nav-index:1;
nav-right:#b2;nav-left:#b4;
nav-down:#b2;nav-up:#b4;
}

button#b2
{
top:40%;left:50%;
nav-index:2;
nav-right:#b3;nav-left:#b1;
nav-down:#b3;nav-up:#b1;
}

button#b3
{
top:70%;left:25%;
nav-index:3;
nav-right:#b4;nav-left:#b2;
nav-down:#b4;nav-up:#b2;
}

button#b4
{
top:40%;left:0%;
nav-index:4;
nav-right:#b1;nav-left:#b3;
nav-down:#b1;nav-up:#b3;
}
```

浏览器支持

目前只有 Opera 支持 nav-right 属性。

定义和用法

nav-right 属性规定当使用 nav-right 导航键时，向何处进行导航。

默认值：	**auto**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.navRight="#div2"

语法

```
nav-right: auto|_id_|_target-name_|inherit;
```

值	描述
auto	浏览器决定导航到哪个元素。
id	规定被导航元素的 id。
target-name	规定被导航的目标框架。
inherit	规定应从父元素继承 nav-right 属性的值。

CSS3 nav-up 属性

实例

规定在使用方向键时向何处导航：

```
button#b1
{
top:20%;left:25%;
nav-index:1;
nav-right:#b2;nav-left:#b4;
nav-down:#b2;nav-up:#b4;
}

button#b2
{
top:40%;left:50%;
nav-index:2;
nav-right:#b3;nav-left:#b1;
nav-down:#b3;nav-up:#b1;
}

button#b3
{
top:70%;left:25%;
nav-index:3;
nav-right:#b4;nav-left:#b2;
nav-down:#b4;nav-up:#b2;
}

button#b4
{
top:40%;left:0%;
nav-index:4;
nav-right:#b1;nav-left:#b3;
nav-down:#b1;nav-up:#b3;
}
```

浏览器支持

目前只有 Opera 支持 nav-up 属性。

定义和用法

nav-up 属性规定当使用 nav-up 导航键时，向何处进行导航。

默认值：	**auto**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.navUp="#div2"

语法

```
nav-up: auto|_id_|_target-name_|inherit;
```

值	描述
auto	浏览器决定导航到哪个元素。
id	规定被导航元素的 id。
target-name	规定被导航的目标框架。
inherit	规定应从父元素继承 nav-up 属性的值。

CSS3 outline-offset 属性

实例

规定边框边缘之外 15 像素处的轮廓：

```
div
{
border:2px solid black;
outline:2px solid red;
outline-offset:15px;
}
```

浏览器支持

所有主流浏览器都支持 outline-offset 属性，除了 Internet Explorer。

定义和用法

outline-offset 属性对轮廓进行偏移，并在边框边缘进行绘制。

轮廓在两方面与边框不同：

- 轮廓不占用空间
- 轮廓可能是非矩形

默认值：	0
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.outlineOffset="15px"

语法

```
outline-offset: _length_|inherit;
```

值	描述
length	轮廓与边框边缘的距离。
inherit	规定应从父元素继承 outline-offset 属性的值。

CSS3 resize 属性

实例

规定可以由用户调整 div 元素的大小：

```
div
{
resize:both;
overflow:auto;
}
```

浏览器支持

Firefox 4+、Chrome 以及 Safari 不支持 resize。

定义和用法

resize 属性规定是否可由用户调整元素的尺寸。

注释：如果希望此属性生效，需要设置元素的 overflow 属性，值可以是 auto、hidden 或 scroll。

默认值：	**none**
继承性：	no
版本：	CSS3
JavaScript 语法：	*object*.style.resize="both"

语法

```
resize: none|both|horizontal|vertical;
```

值	描述
none	用户无法调整元素的尺寸。
both	用户可调整元素的高度和宽度。
horizontal	用户可调整元素的宽度。
vertical	用户可调整元素的高度。

相关页面

CSS3 教程：CSS3 用户界面

CSS 选择器参考手册

我们会定期对 W3School 的 CSS 参考手册进行浏览器测试。

CSS3 选择器

在 CSS 中，选择器是一种模式，用于选择需要添加样式的元素。

"CSS" 列指示该属性是在哪个 CSS 版本中定义的。（CSS1、CSS2 还是 CSS3。）

选择器	例子	例子描述	CSS
.class	.intro	选择 class="intro" 的所有元素。	1
#id	#firstname	选择 id="firstname" 的所有元素。	1
*	*	选择所有元素。	2
element	p	选择所有 <p> 元素。	1
element,element	div,p	选择所有 <div> 元素和所有 <p> 元素。	1
element element	div p	选择 <div> 元素内部的所有 <p> 元素。	1
element>element	div>p	选择父元素为 <div> 元素的所有 <p> 元素。	2
element+element	div+p	选择紧接在 <div> 元素之后的所有 <p> 元素。	2
[*attribute*]	[target]	选择带有 target 属性所有元素。	2
[*attribute=value*]	[target=_blank]	选择 target="_blank" 的所有元素。	2
[*attribute~=value*]	[title~=flower]	选择 title 属性包含单词 "flower" 的所有元素。	2
[*attribute|=value*]	[lang|=en]	选择 lang 属性值以 "en" 开头的所有元素。	2
:link	a:link	选择所有未被访问的链接。	1
:visited	a:visited	选择所有已被访问的链接。	1
:active	a:active	选择活动链接。	1
:hover	a:hover	选择鼠标指针位于其上的链接。	1
:focus	input:focus	选择获得焦点的 input 元素。	2
:first-letter	p:first-letter	选择每个 <p> 元素的首字母。	1

:first-line	p:first-line	选择每个 <p> 元素的首行。	1
:first-child	p:first-child	选择属于父元素的第一个子元素的每个 <p> 元素。	2
:before	p:before	在每个 <p> 元素的内容之前插入内容。	2
:after	p:after	在每个 <p> 元素的内容之后插入内容。	2
:lang(*language*)	p:lang(it)	选择带有以 "it" 开头的 lang 属性值的每个 <p> 元素。	2
element1~element2	p~ul	选择前面有 <p> 元素的每个 元素。	3
[*attribute*^=*value*]	a[src^="https"]	选择其 src 属性值以 "https" 开头的每个 <a> 元素。	3
[*attribute*$=*value*]	a[src$=".pdf"]	选择其 src 属性以 ".pdf" 结尾的所有 <a> 元素。	3
[*attribute**=*value*]	a[src*="abc"]	选择其 src 属性中包含 "abc" 子串的每个 <a> 元素。	3
:first-of-type	p:first-of-type	选择属于其父元素的首个 <p> 元素的每个 <p> 元素。	3
:last-of-type	p:last-of-type	选择属于其父元素的最后 <p> 元素的每个 <p> 元素。	3
:only-of-type	p:only-of-type	选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。	3
:only-child	p:only-child	选择属于其父元素的唯一子元素的每个 <p> 元素。	3
:nth-child(*n*)	p:nth-child(2)	选择属于其父元素的第二个子元素的每个 <p> 元素。	3
:nth-last-child(*n*)	p:nth-last-child(2)	同上，从最后一个子元素开始计数。	3
:nth-of-type(*n*)	p:nth-of-type(2)	选择属于其父元素第二个 <p> 元素的每个 <p> 元素。	3
:nth-last-of-type(*n*)	p:nth-last-of-type(2)	同上，但是从最后一个子元素开始计数。	3
:last-child	p:last-child	选择属于其父元素最后一个子元素每个 <p> 元素。	3
:root	:root	选择文档的根元素。	3
:empty	p:empty	选择没有子元素的每个 <p> 元素（包括文本节点）。	3
:target	#news:target	选择当前活动的 #news 元素。	3

:enabled	input:enabled	选择每个启用的 <input> 元素。	3
:disabled	input:disabled	选择每个禁用的 <input> 元素	3
:checked	input:checked	选择每个被选中的 <input> 元素。	3
:not(*selector*)	:not(p)	选择非 <p> 元素的每个元素。	3
::selection	::selection	选择被用户选取的元素部分。	3

CSS 听觉参考手册

听觉样式表

听觉样式表可把语音合成与音响效果相组合，使用户可以听到信息，而无需进行阅读。

听觉呈现可用于：

- 视觉能力低弱的人士
- 帮助用户学习阅读
- 帮助有阅读障碍的用户
- 家庭娱乐
- 在汽车中使用

听觉呈现通常会把文档转化为纯文本，然后传给屏幕阅读器（可读出屏幕上所有字符的一种程序）。

听觉样式表的一个例子：

```
h1, h2, h3, h4
{
voice-family: male;
richness: 80;
cue-before: url("beep.au")
}
```

上面的例子可以让语音合成器演奏一段声音，然后用男性的声音读出标题。

CSS2 听觉参考

W3C: "W3C" 列的数字显示出属性由哪个 CSS 标准定义（CSS1 还是 CSS2）。

属性	描述	值	W3C
azimuth	Sets where the sound/voices should come from (horizontally)	`angle` `left-side` `far-left` `left` `center-left` `center` `center-right` `right` `far-right` `right-side` `behind` `leftwards` `rightwards`	2
cue	A shorthand property for setting the cue-before and cue-after properties in one declaration	`cue-before` `cue-after`	2

cue-after	Specifies a sound to be played after speaking an element's content to delimit it from other	none url	2
cue-before	Specifies a sound to be played before speaking an element's content to delimit it from other	none url	2
elevation	Sets where the sound/voices should come from (vertically)	angle belo level above higher lower	2
pause	A shorthand property for setting the pause-before and pause-after properties in one declaration	pause-before pause-after	2
pause-after	Specifies a pause after speaking an element's content	time %	2
pause-before	Specifies a pause before speaking an element's content	time %	2
pitch	Specifies the speaking voice	frequency x-low low medium high x-high	2
pitch-range	Specifies the variation in the speaking voice. (Monotone voice or animated voice?)	number	2
play-during	Specifies a sound to be played while speaking an element's content	auto none url mix repeat	2
richness	Specifies the richness in the speaking voice. (Rich voice or thin voice?)	number	2
speak	Specifies whether content will render aurally	normal none spell-out	2
speak-header	Specifies how to handle table headers. Should the headers be spoken before every cell, or only before a cell with a different header than the previous cell	always once	2
speak-numeral	Specifies how to speak numbers	digits continuous	2
speak-punctuation	Specifies how to speak punctuation characters	none code	2
speech-rate	Specifies the speed of the speaking	number x-slow slow medium fast x-fast faster slower	2
	Specifies the "stress" in the		

stress	speaking voice	`number`	2
voice-family	A prioritized list of voice family names that contain specific voices	`specific-voice` `generic-voice`	2
volume	Specifies the volume of the speaking	`number` `%` `silent` `x-soft` `soft` `medium` `loud` `x-loud`	2

CSS 网络安全字体组合

常用的字体组合

font-family 属性应该使用若干种字体名称作为回退系统，以确保浏览器/操作系统之间的最大兼容性。如果浏览器不支持第一个字体，则会尝试下一个。

请以您喜欢的字体开始，并以通用字体系列结束，以便使浏览器在通用系统中挑选相似的字体，如果没有其他字体可用的话：

实例

```
p{font-family:'Times New Roman', Times, serif}
```

下面是最常用的字体组合，根据通用系统进行汇总：

Serif 字体

font-family	示例文本
Georgia, serif	**This is a heading** This is a paragraph
'Palatino Linotype', 'Book Antiqua', Palatino, serif	**This is a heading** This is a paragraph
'Times New Roman', Times, serif	**This is a heading** This is a paragraph

---

Sans-Serif 字体

font-family	示例文本
Arial, Helvetica, sans-serif	**This is a heading** This is a paragraph
'Arial Black', Gadget, sans-serif	**This is a heading** **This is a paragraph**
'Comic Sans MS', cursive, sans-serif	**This is a heading** This is a paragraph
Impact, Charcoal, sans-serif	**This is a heading** **This is a paragraph**
'Lucida Sans Unicode', 'Lucida Grande', sans-serif	**This is a heading** This is a paragraph
Tahoma, Geneva, sans-serif	**This is a heading** This is a paragraph
'Trebuchet MS', Helvetica, sans-serif	**This is a heading** This is a paragraph
Verdana, Geneva, sans-serif	**This is a heading** This is a paragraph

---

Monospace 字体

font-family	示例文本
'Courier New', Courier, monospace	**This is a heading** This is a paragraph
'Lucida Console', Monaco, monospace	**This is a heading** This is a paragraph

CSS 单位

尺寸

单位	描述
%	百分比
in	英寸
cm	厘米
mm	毫米
em	1em 等于当前的字体尺寸。2em 等于当前字体尺寸的两倍。例如，如果某元素以 12pt 显示，那么 2em 是24pt。在 CSS 中，em 是非常有用的单位，因为它可以自动适应用户所使用的字体。
ex	一个 ex 是一个字体的 x-height。 (x-height 通常是字体尺寸的一半。)
pt	磅 (1 pt 等于 1/72 英寸)
pc	12 点活字 (1 pc 等于 12 点)
px	像素 (计算机屏幕上的一个点)

颜色

单位	描述
(颜色名)	颜色名称 (比如 red)
rgb(x,x,x)	RGB 值 (比如 rgb(255,0,0))
rgb(x%, x%, x%)	RGB 百分比值 (比如 rgb(100%,0%,0%))
#rrggbb	十六进制数 (比如 #ff0000)

CSS 颜色

颜色是通过对红、绿和蓝光的组合来显示的。

颜色值

CSS 颜色使用组合了红绿蓝颜色值 (RGB) 的十六进制 (hex) 表示法进行定义。对光源进行设置的最低值可以是 0（十六进制 00）。最高值是 255（十六进制 FF）。

十六进制值使用三个双位数来编写，并以 # 符号开头。

颜色	颜色 HEX	颜色 RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

1600 万种不同的颜色

从 0 到 255 种红绿蓝值能够组合出总共超过一千六百万种不同的颜色（根据 256 x 256 x 256 计算）。

大多数现代的显示器都能显示出至少 16384 种不同的颜色。

如果您查看下面的表格，您将看到红光从 0 到 255 变化后的结果，此时绿光和蓝光为零。

如需查看红光由 0 向 255 变化的完整颜色混合器列表，请点击下面的十六进制值或 rgb 值。

Red Light	HEX	RGB
	#000000	rgb(0,0,0)
	#080000	rgb(8,0,0)
	#100000	rgb(16,0,0)

	#180000	rgb(24,0,0)
	#200000	rgb(32,0,0)
	#280000	rgb(40,0,0)
	#300000	rgb(48,0,0)
	#380000	rgb(56,0,0)
	#400000	rgb(64,0,0)
	#480000	rgb(72,0,0)
	#500000	rgb(80,0,0)
	#580000	rgb(88,0,0)
	#600000	rgb(96,0,0)
	#680000	rgb(104,0,0)
	#700000	rgb(112,0,0)
	#780000	rgb(120,0,0)
	#800000	rgb(128,0,0)
	#880000	rgb(136,0,0)
	#900000	rgb(144,0,0)
	#980000	rgb(152,0,0)
	#A00000	rgb(160,0,0)
	#A80000	rgb(168,0,0)
	#B00000	rgb(176,0,0)
	#B80000	rgb(184,0,0)
	#C00000	rgb(192,0,0)
	#C80000	rgb(200,0,0)
	#D00000	rgb(208,0,0)
	#D80000	rgb(216,0,0)
	#E00000	rgb(224,0,0)
	#E80000	rgb(232,0,0)
	#F00000	rgb(240,0,0)
	#F80000	rgb(248,0,0)
	#FF0000	rgb(255,0,0)

灰阶

灰色使用所有光源的等量的光线来显示。为了让您更方便地选择正确的灰色，我们已经为您编辑了一个灰色列表：

灰阶	HEX	RGB
	#000000	rgb(0,0,0)
	#080808	rgb(8,8,8)
	#101010	rgb(16,16,16)
	#181818	rgb(24,24,24)
	#202020	rgb(32,32,32)
	#282828	rgb(40,40,40)
	#303030	rgb(48,48,48)
	#383838	rgb(56,56,56)
	#404040	rgb(64,64,64)
	#484848	rgb(72,72,72)
	#505050	rgb(80,80,80)
	#585858	rgb(88,88,88)
	#606060	rgb(96,96,96)
	#686868	rgb(104,104,104)
	#707070	rgb(112,112,112)
	#787878	rgb(120,120,120)
	#808080	rgb(128,128,128)
	#888888	rgb(136,136,136)
	#909090	rgb(144,144,144)
	#989898	rgb(152,152,152)
	#A0A0A0	rgb(160,160,160)
	#A8A8A8	rgb(168,168,168)
	#B0B0B0	rgb(176,176,176)
	#B8B8B8	rgb(184,184,184)
	#C0C0C0	rgb(192,192,192)
	#C8C8C8	rgb(200,200,200)
	#D0D0D0	rgb(208,208,208)

	#D8D8D8	rgb(216,216,216)
	#E0E0E0	rgb(224,224,224)
	#E8E8E8	rgb(232,232,232)
	#F0F0F0	rgb(240,240,240)
	#F8F8F8	rgb(248,248,248)
	#FFFFFF	rgb(255,255,255)

网络安全色？

多年前，当电脑只支持最多 256 种颜色时，216 种“网络安全色”列表被定义为 Web 标准，并保留了 40 种固定的系统颜色。

现在，这些都不重要了，因为大多数电脑都能显示数百万种颜色，但是选择权依然在您手里。

这个 216 跨浏览器调色板被创建的目的是确保当计算机运行 256 色调色板时能够正确地显示颜色：

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF
669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF

66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF
FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF

CSS 合法颜色值

CSS 颜色

可以用以下方法来规定 CSS 中的颜色：

- 十六进制色
- RGB 颜色
- RGBA 颜色
- HSL 颜色
- HSLA 颜色
- 预定义/跨浏览器颜色名

十六进制颜色

所有浏览器都支持十六进制颜色值。

十六进制颜色是这样规定的：#RRGGBB，其中的 RR（红色）、GG（绿色）、BB（蓝色）十六进制整数规定了颜色的成分。所有值必须介于 0 与 FF 之间。

举例说，#0000ff 值显示为蓝色，这是因为蓝色成分被设置为最高值（ff），而其他成分被设置为 0。

实例

```
p
{
background-color:#0000ff;
}
```

RGB 颜色

所有浏览器都支持 RGB 颜色值。

RGB 颜色值是这样规定的：rgb(red, green, blue)。每个参数 (red、green 以及 blue) 定义颜色的强度，可以是介于 0 与 255 之间的整数，或者是百分比值（从 0% 到 100%）。

举例说，rgb(0,0,255) 值显示为蓝色，这是因为 blue 参数被设置为最高值（255），而其他被设置为 0。

同样地，下面的值定义了相同的颜色：rgb(0,0,255) 和 rgb(0%,0%,100%)。

实例

```
p
{
background-color:rgb(255,0,0);
}
```

RGBA 颜色

RGBA 颜色值得到以下浏览器的支持：IE9+、Firefox 3+、Chrome、Safari 以及 Opera 10+。

RGBA 颜色值是 RGB 颜色值的扩展，带有一个 alpha 通道 - 它规定了对象的不透明度。

RGBA 颜色值是这样规定的：rgba(red, green, blue, alpha)。alpha 参数是介于 0.0（完全透明）与 1.0（完全不透明）的数字。

实例

```
p
{
background-color:rgba(255,0,0,0.5);
}
```

HSL 颜色

HSL 颜色值得到以下浏览器的支持：IE9+、Firefox、Chrome、Safari 以及 Opera 10+。

HSL 指的是 hue（色调）、saturation（饱和度）、lightness（亮度） - 表示颜色柱面坐标表示法。

HSL 颜色值是这样规定的：hsl(hue, saturation, lightness)。

Hue 是色盘上的度数（从 0 到 360） - 0 (或 360) 是红色，120 是绿色，240 是蓝色。Saturation 是百分比值；0% 意味着灰色，而 100% 是全彩。Lightness 同样是百分比值；0% 是黑色，100% 是白色。

实例

```
p
{
background-color:hsl(120,65%,75%);
}
```

HSLA 颜色

HSLA 颜色值得到以下浏览器的支持：IE9+、Firefox 3+、Chrome、Safari 以及 Opera 10+。

HSLA 颜色值是 HSL 颜色值的扩展，带有一个 alpha 通道 - 它规定了对象的不透明度。

HSLA 颜色值是这样规定的：hsla(hue, saturation, lightness, alpha)，其中的 alpha 参数定义不透明度。alpha 参数是介于 0.0（完全透明）与 1.0（完全不透明）的数字。

实例

```
p
{
background-color:hsla(120,65%,75%,0.3);
}
```

CSS 颜色名

CSS 颜色名

所有浏览器都支持的颜色名。

HTML 和 CSS 颜色规范中定义了 147 中颜色名（17 种标准颜色加 130 种其他颜色）。下面的表格中列出了所有这些颜色，以及它们的十六进制值。

提示：17 种标准色是 aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow。

点击颜色名（或十六进制值）来查看这种颜色作为背景色搭配不同的文本颜色：

以颜色名排序

根据十六进制值排序的相同列表

颜色名	十六进制颜色值	颜色
AliceBlue	#F0F8FF	
AntiqueWhite	#FAEBD7	
Aqua	#00FFFF	
Aquamarine	#7FFFD4	
Azure	#F0FFFF	
Beige	#F5F5DC	
Bisque	#FFE4C4	
Black	#000000	
BlanchedAlmond	#FFEBCD	
Blue	#0000FF	
BlueViolet	#8A2BE2	
Brown	#A52A2A	
BurlyWood	#DEB887	
CadetBlue	#5F9EA0	
Chartreuse	#7FFF00	
Chocolate	#D2691E	

Coral	#FF7F50
CornflowerBlue	#6495ED
Cornsilk	#FFF8DC
Crimson	#DC143C
Cyan	#00FFFF
DarkBlue	#00008B
DarkCyan	#008B8B
DarkGoldenRod	#B8860B
DarkGray	#A9A9A9
DarkGreen	#006400
DarkKhaki	#BDB76B
DarkMagenta	#8B008B
DarkOliveGreen	#556B2F
Darkorange	#FF8C00
DarkOrchid	#9932CC
DarkRed	#8B0000
DarkSalmon	#E9967A
DarkSeaGreen	#8FBC8F
DarkSlateBlue	#483D8B
DarkSlateGray	#2F4F4F
DarkTurquoise	#00CED1
DarkViolet	#9400D3
DeepPink	#FF1493
DeepSkyBlue	#00BFFF
DimGray	#696969
DodgerBlue	#1E90FF
Feldspar	#D19275
FireBrick	#B22222
FloralWhite	#FFFAF0
ForestGreen	#228B22
Fuchsia	#FF00FF
Gainsboro	#DCDCDC

GhostWhite	#F8F8FF
Gold	#FFD700
GoldenRod	#DAA520
Gray	#808080
Green	#008000
GreenYellow	#ADFF2F
HoneyDew	#F0FFF0
HotPink	#FF69B4
IndianRed	#CD5C5C
Indigo	#4B0082
Ivory	#FFFFF0
Khaki	#F0E68C
Lavender	#E6E6FA
LavenderBlush	#FFF0F5
LawnGreen	#7CFC00
LemonChiffon	#FFFACD
LightBlue	#ADD8E6
LightCoral	#F08080
LightCyan	#E0FFFF
LightGoldenRodYellow	#FAFAD2
LightGrey	#D3D3D3
LightGreen	#90EE90
LightPink	#FFB6C1
LightSalmon	#FFA07A
LightSeaGreen	#20B2AA
LightSkyBlue	#87CEFA
LightSlateBlue	#8470FF
LightSlateGray	#778899
LightSteelBlue	#B0C4DE
LightYellow	#FFFFE0
Lime	#00FF00

LimeGreen	#32CD32
Linen	#FAF0E6
Magenta	#FF00FF
Maroon	#800000
MediumAquaMarine	#66CDAA
MediumBlue	#0000CD
MediumOrchid	#BA55D3
MediumPurple	#9370D8
MediumSeaGreen	#3CB371
MediumSlateBlue	#7B68EE
MediumSpringGreen	#00FA9A
MediumTurquoise	#48D1CC
MediumVioletRed	#C71585
MidnightBlue	#191970
MintCream	#F5FFFA
MistyRose	#FFE4E1
Moccasin	#FFE4B5
NavajoWhite	#FFDEAD
Navy	#000080
OldLace	#FDF5E6
Olive	#808000
OliveDrab	#6B8E23
Orange	#FFA500
OrangeRed	#FF4500
Orchid	#DA70D6
PaleGoldenRod	#EEE8AA
PaleGreen	#98FB98
PaleTurquoise	#AFEEEE
PaleVioletRed	#D87093
PapayaWhip	#FFEFD5
PeachPuff	#FFDAB9
Peru	#CD853F

Pink	#FFC0CB
Plum	#DDA0DD
PowderBlue	#B0E0E6
Purple	#800080
Red	#FF0000
RosyBrown	#BC8F8F
RoyalBlue	#4169E1
SaddleBrown	#8B4513
Salmon	#FA8072
SandyBrown	#F4A460
SeaGreen	#2E8B57
SeaShell	#FFF5EE
Sienna	#A0522D
Silver	#C0C0C0
SkyBlue	#87CEEB
SlateBlue	#6A5ACD
SlateGray	#708090
Snow	#FFFAFA
SpringGreen	#00FF7F
SteelBlue	#4682B4
Tan	#D2B48C
Teal	#008080
Thistle	#D8BFD8
Tomato	#FF6347
Turquoise	#40E0D0
Violet	#EE82EE
VioletRed	#D02090
Wheat	#F5DEB3
White	#FFFFFF
WhiteSmoke	#F5F5F5
Yellow	#FFFF00

YellowGreen	#9ACD32

CSS 颜色十六进制值

根据十六进制值排序

所有浏览器都支持的颜色名

根据颜色名排序的相同列表

颜色名	十六进制颜色值	颜色
Black	#000000	
Navy	#000080	
DarkBlue	#00008B	
MediumBlue	#0000CD	
Blue	#0000FF	
DarkGreen	#006400	
Green	#008000	
Teal	#008080	
DarkCyan	#008B8B	
DeepSkyBlue	#00BFFF	
DarkTurquoise	#00CED1	
MediumSpringGreen	#00FA9A	
Lime	#00FF00	
SpringGreen	#00FF7F	
Aqua	#00FFFF	
Cyan	#00FFFF	
MidnightBlue	#191970	
DodgerBlue	#1E90FF	
LightSeaGreen	#20B2AA	
ForestGreen	#228B22	
SeaGreen	#2E8B57	
DarkSlateGray	#2F4F4F	
LimeGreen	#32CD32	

MediumSeaGreen	#3CB371
Turquoise	#40E0D0
RoyalBlue	#4169E1
SteelBlue	#4682B4
DarkSlateBlue	#483D8B
MediumTurquoise	#48D1CC
Indigo	#4B0082
DarkOliveGreen	#556B2F
CadetBlue	#5F9EA0
CornflowerBlue	#6495ED
MediumAquaMarine	#66CDAA
DimGray	#696969
DimGrey	#696969
SlateBlue	#6A5ACD
OliveDrab	#6B8E23
SlateGray	#708090
LightSlateGray	#778899
MediumSlateBlue	#7B68EE
LawnGreen	#7CFC00
Chartreuse	#7FFF00
Aquamarine	#7FFFD4
Maroon	#800000
Purple	#800080
Olive	#808000
Gray	#808080
SkyBlue	#87CEEB
LightSkyBlue	#87CEFA
BlueViolet	#8A2BE2
DarkRed	#8B0000
DarkMagenta	#8B008B
SaddleBrown	#8B4513
DarkSeaGreen	#8FBC8F

LightGreen	#90EE90
MediumPurple	#9370DB
DarkViolet	#9400D3
PaleGreen	#98FB98
DarkOrchid	#9932CC
YellowGreen	#9ACD32
Sienna	#A0522D
Brown	#A52A2A
DarkGray	#A9A9A9
LightBlue	#ADD8E6
GreenYellow	#ADFF2F
PaleTurquoise	#AFEEEE
LightSteelBlue	#B0C4DE
PowderBlue	#B0E0E6
FireBrick	#B22222
DarkGoldenRod	#B8860B
MediumOrchid	#BA55D3
RosyBrown	#BC8F8F
DarkKhaki	#BDB76B
Silver	#C0C0C0
MediumVioletRed	#C71585
IndianRed	#CD5C5C
Peru	#CD853F
Chocolate	#D2691E
Tan	#D2B48C
LightGray	#D3D3D3
Thistle	#D8BFD8
Orchid	#DA70D6
GoldenRod	#DAA520
PaleVioletRed	#DB7093
Crimson	#DC143C

Gainsboro	#DCDCDC
Plum	#DDA0DD
BurlyWood	#DEB887
LightCyan	#E0FFFF
Lavender	#E6E6FA
DarkSalmon	#E9967A
Violet	#EE82EE
PaleGoldenRod	#EEE8AA
LightCoral	#F08080
Khaki	#F0E68C
AliceBlue	#F0F8FF
HoneyDew	#F0FFF0
Azure	#F0FFFF
SandyBrown	#F4A460
Wheat	#F5DEB3
Beige	#F5F5DC
WhiteSmoke	#F5F5F5
MintCream	#F5FFFA
GhostWhite	#F8F8FF
Salmon	#FA8072
AntiqueWhite	#FAEBD7
Linen	#FAF0E6
LightGoldenRodYellow	#FAFAD2
OldLace	#FDF5E6
Red	#FF0000
Fuchsia	#FF00FF
Magenta	#FF00FF
DeepPink	#FF1493
OrangeRed	#FF4500
Tomato	#FF6347
HotPink	#FF69B4
Coral	#FF7F50

Darkorange	#FF8C00
LightSalmon	#FFA07A
Orange	#FFA500
LightPink	#FFB6C1
Pink	#FFC0CB
Gold	#FFD700
PeachPuff	#FFDAB9
NavajoWhite	#FFDEAD
Moccasin	#FFE4B5
Bisque	#FFE4C4
MistyRose	#FFE4E1
BlanchedAlmond	#FFEBCD
PapayaWhip	#FFEFD5
LavenderBlush	#FFF0F5
SeaShell	#FFF5EE
Cornsilk	#FFF8DC
LemonChiffon	#FFFACD
FloralWhite	#FFFAF0
Snow	#FFFAFA
Yellow	#FFFF00
LightYellow	#FFFFE0
Ivory	#FFFFF0
White	#FFFFFF

免责声明

W3School提供的内容仅用于培训。我们不保证内容的正确性。通过使用本站内容随之而来的风险与本站无关。W3School简体中文版的所有内容仅供测试，对任何法律问题及风险不承担任何责任。